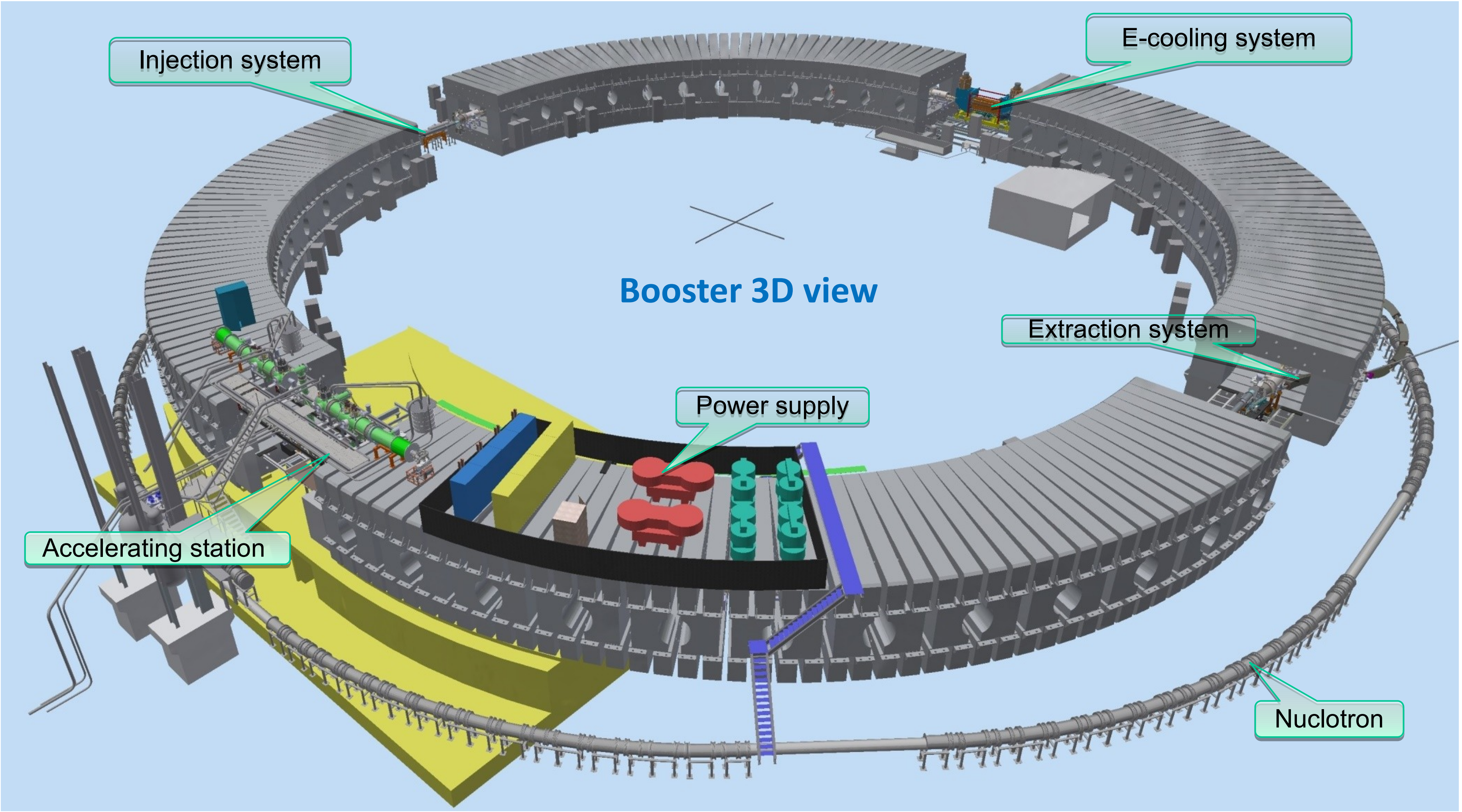
APPLICATION OF QUANTUM TECHNOLOGIES FOR THE DEVELOPMENT OF AN INTELLECTUAL CONTROL SYSTEM TO SETUP CURRENTS OF THE CORRECTIVE MAGNETS FOR THE BOOSTER SYNCHROTRON OF THE NICA FACILITY

D.V. Monakhov*, A.G. Reshetnikov, G.P. Reshetnikov, S.V. Ulyanov

* monahov@sunse.jinr.ru



According to the results of the last month, the expansion of the yoke under the influence of temperature was confirmed. Almost all the distances between the yoke reference holes increased by an average of 100 microns (distance 4 m) with an increase in temperature (air) by 2-3 degrees. Which corresponds to the coefficient of linear expansion of steel.

Tasks for Booster correction system:

- First Turn Correction
- Closed orbit correction
- Betatron Frequency Spread Correction
- Betatron resonance correction

*At the stage of commissioning for the correction of the magnetic field, it is planned to place **24 dipole** (**normal** and **skew**) correctors in the Booster for correcting the horizontal and vertical projections of a closed orbit and **8 multipole** correctors for correcting betatron resonances: Qx - Qy = 0; Qx - 2Qy = -5; 3Qx = 14.*
Each dipole corrector contains two groups of windings and a multipole corrector contains four groups of windings

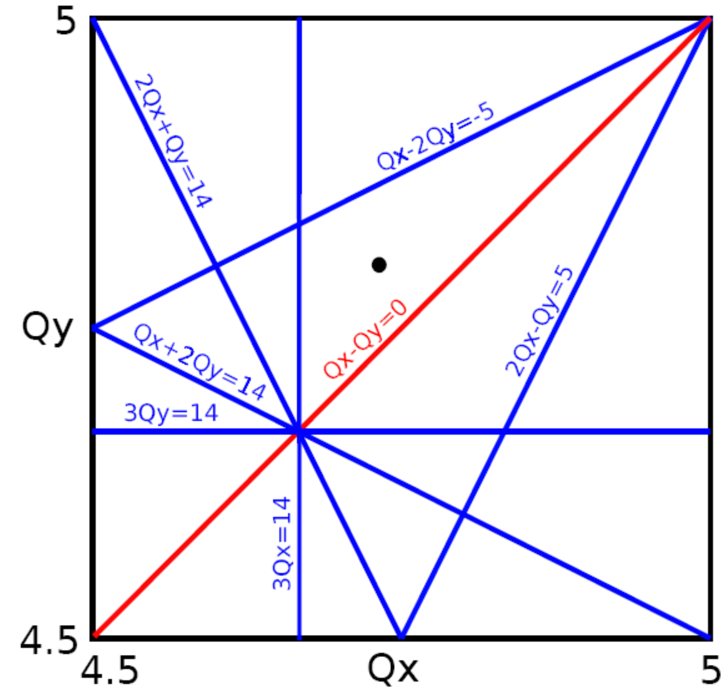
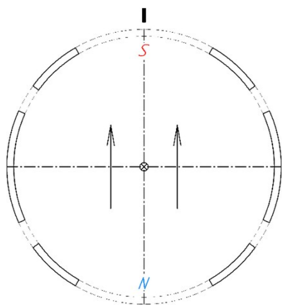
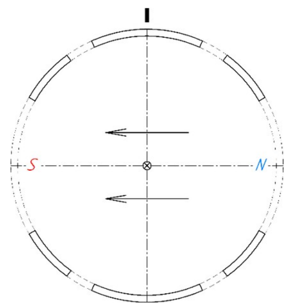


Diagram of resonances

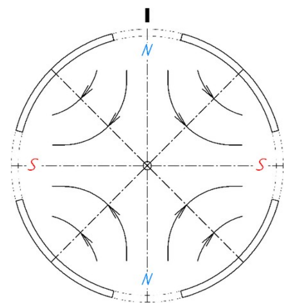
Booster have 32 correctors



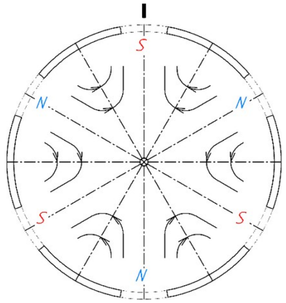
“Normal”
Dipole



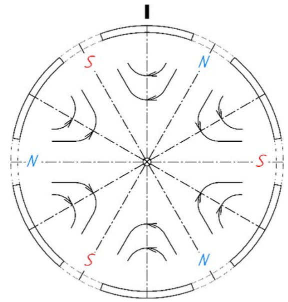
“Skew”
Dipole



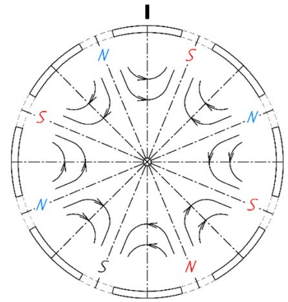
“Skew”
Quadrupole



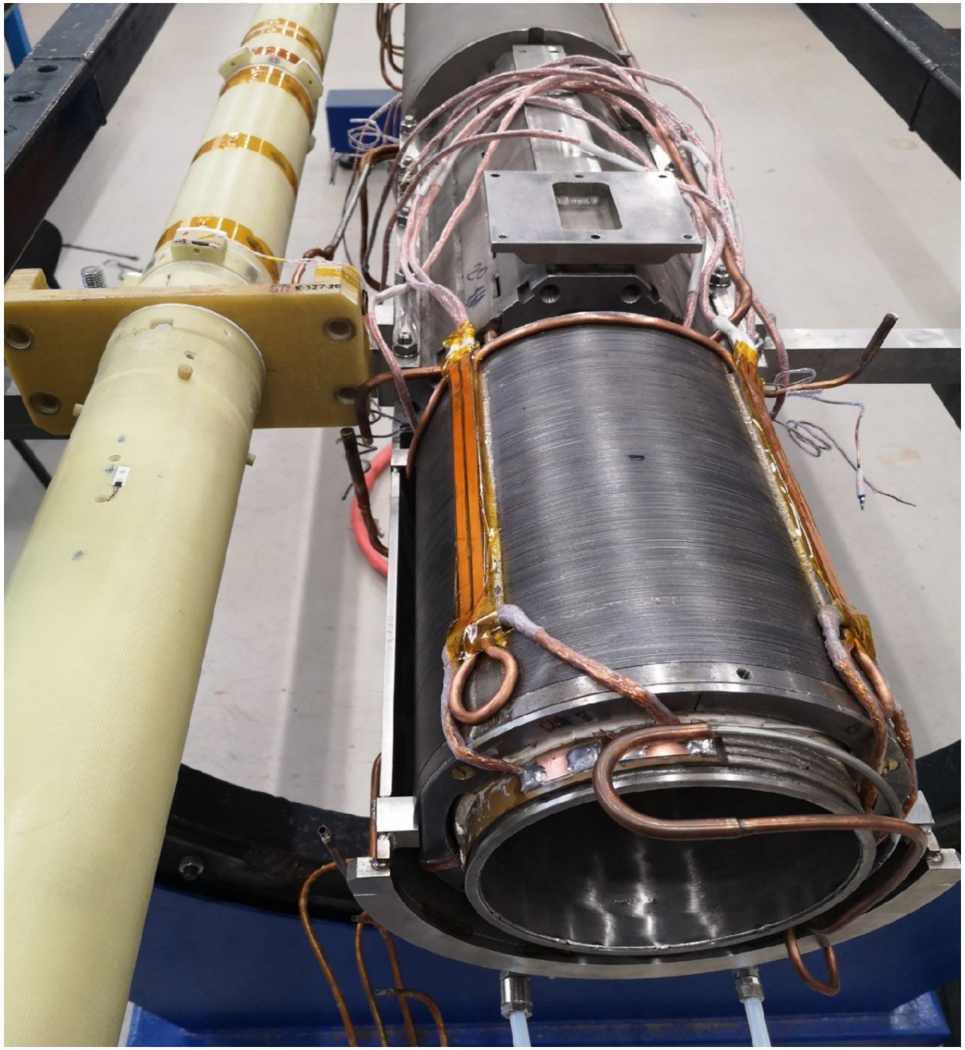
“Normal”
Sextupole



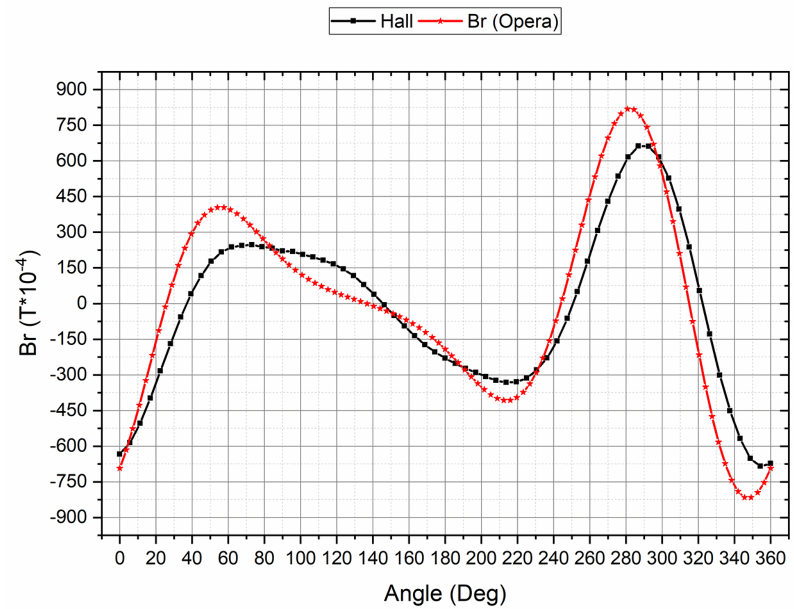
“Skew”
Sextupole



“Normal”
Octupole



Corrective magnets – they are not ideal!



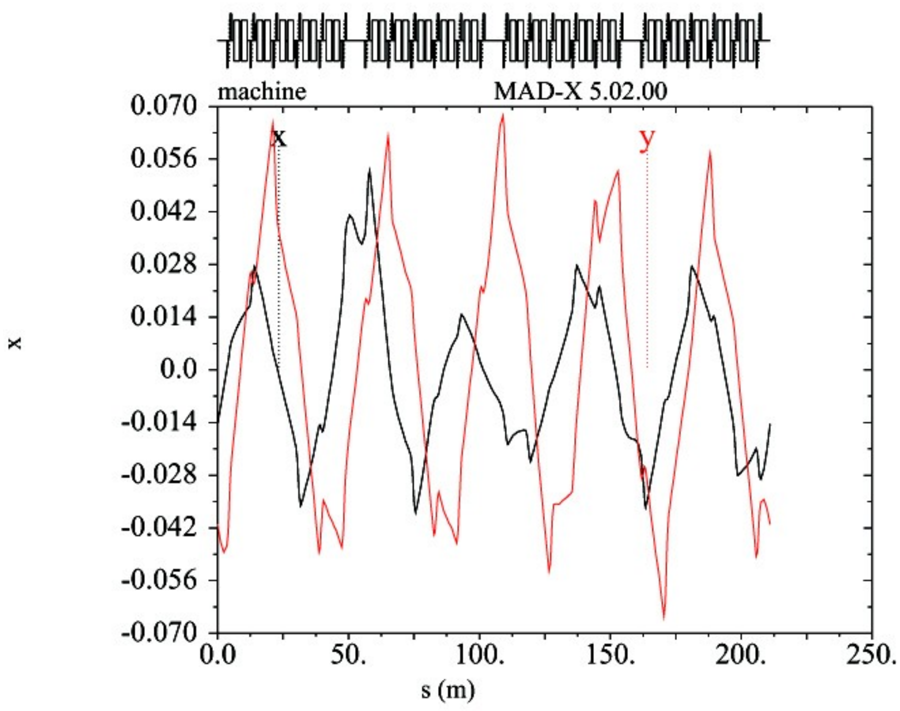
Field Type	Measurements	FEM 2D
Quadrupole (a_1)	0.03834	0.04422
Sextupole (b_2)	0.02111	0.02548
Sextupole (a_2)	0.01028	0.02501
Octupole (b_3)	7.766*10^{-3}	8.446*10^{-3}

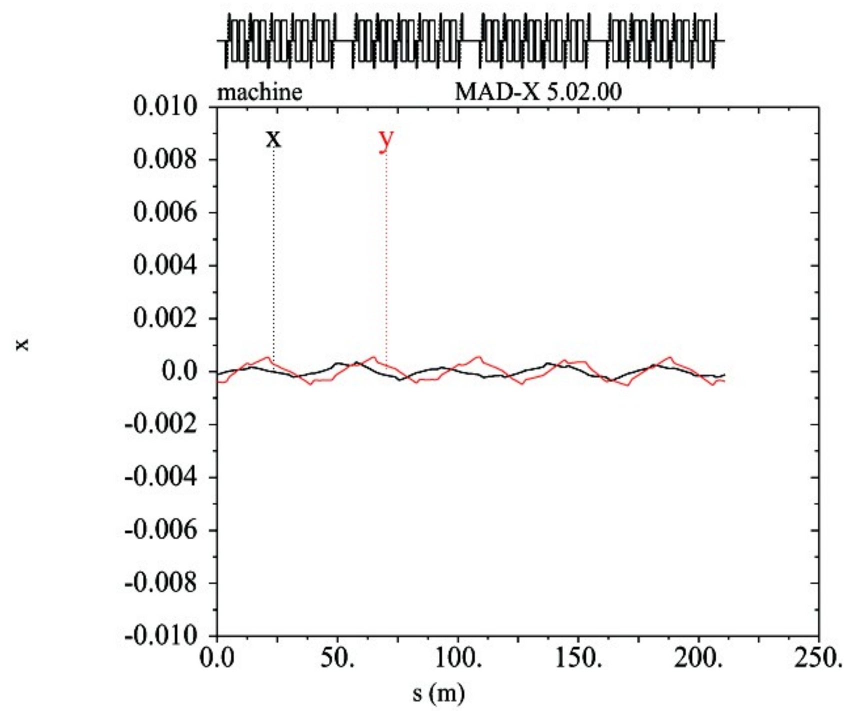
Field Type	Measurements	FEM 2D
Quadrupole (a_1)	0.03834	0.04422
Sextupole (b_2)	0.02111	0.02548
Sextupole (a_2)	0.01028	0.02501
Octupole (b_3)	7.766*10^{-3}	8.446*10^{-3}

Field Type	Measurements	FEM 2D
Quadrupole (a_1)	0.03834	0.04422
Sextupole (b_2)	0.02111	0.02548
Sextupole (a_2)	0.01028	0.02501
Octupole (b_3)	7.766*10^{-3}	8.446*10^{-3}

Field Type	Measurements	FEM 2D
Quadrupole (a_1)	0.03834	0.04422
Sextupole (b_2)	0.02111	0.02548
Sextupole (a_2)	0.01028	0.02501
Octupole (b_3)	7.766*10^{-3}	8.446*10^{-3}

	FEM 2D
0.03834	0.04422
0.02111	0.02548
0.01028	0.02501
7.766*10^{-3}	8.446*10^{-3}

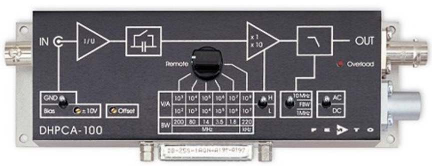




For the selected working point (Qx = 4.8 and Qy = 4.85) the total value of the frequency shifts should not exceed ΔQx,y <0.05.

Q-measurement system tested at Nuclotron

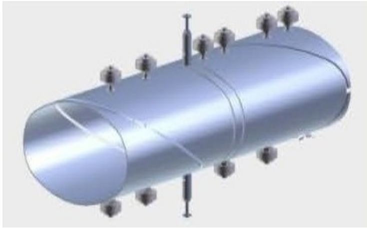
Variable gain amplifiers

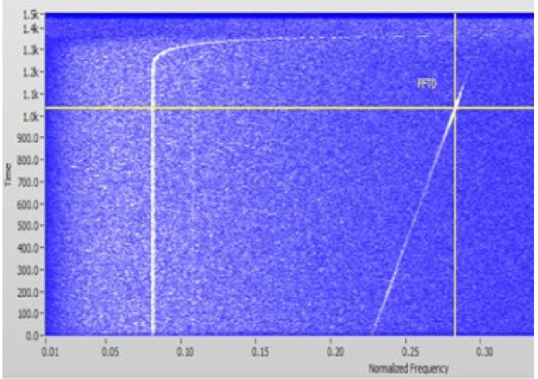


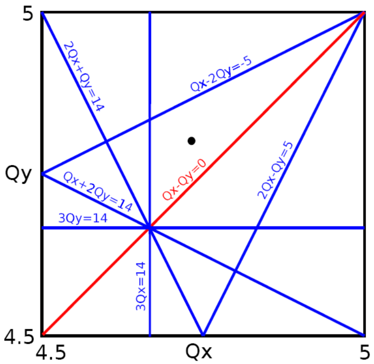
NI PXIe-1082 crate with FlexRIO modules

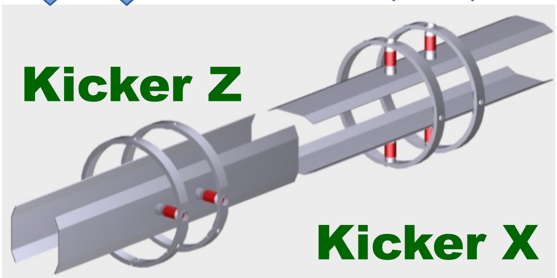
RF Amplifier AR 800A3A

Impedance transformers







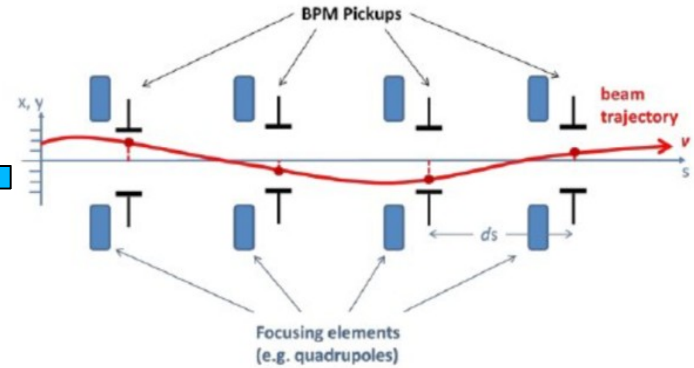


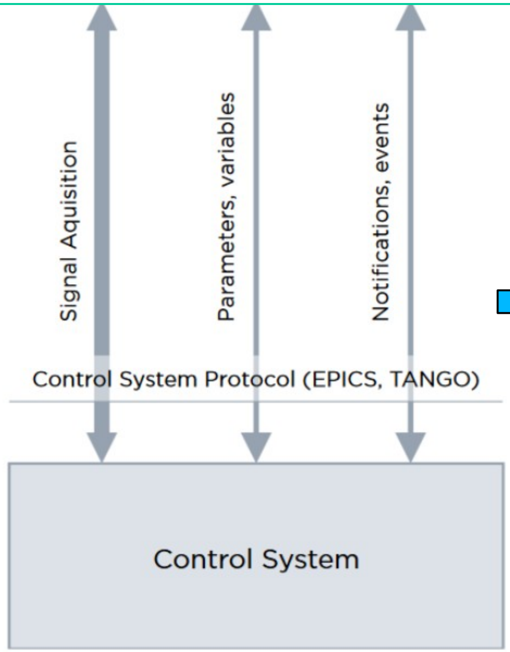
Pick-up electrodes X and Z

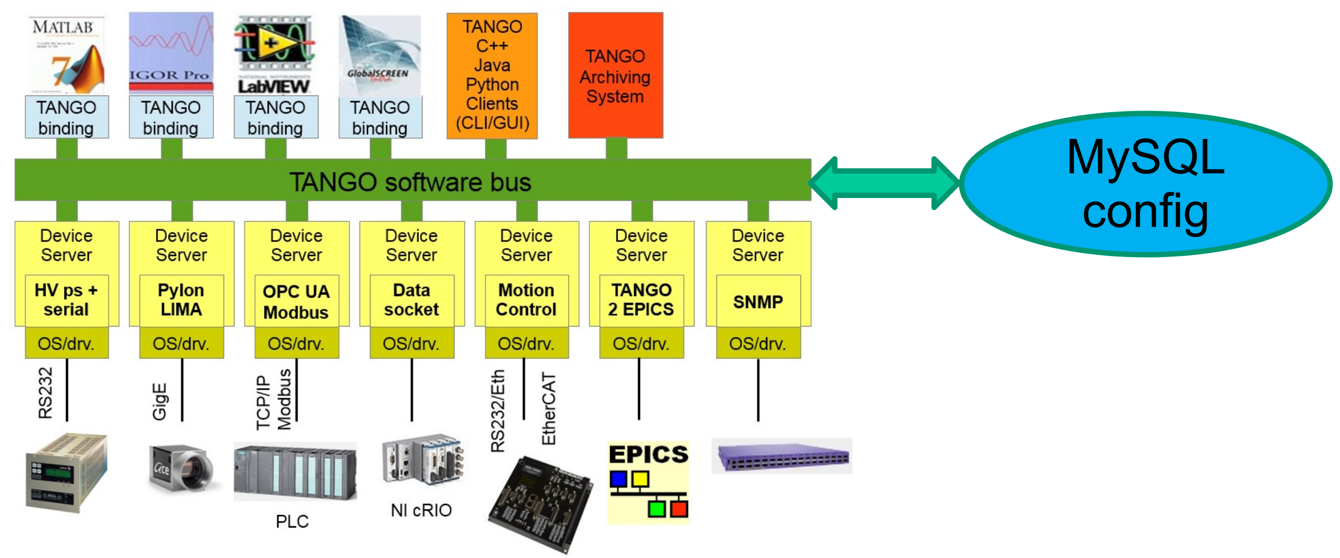
Libera BPM system

Libera Hadron

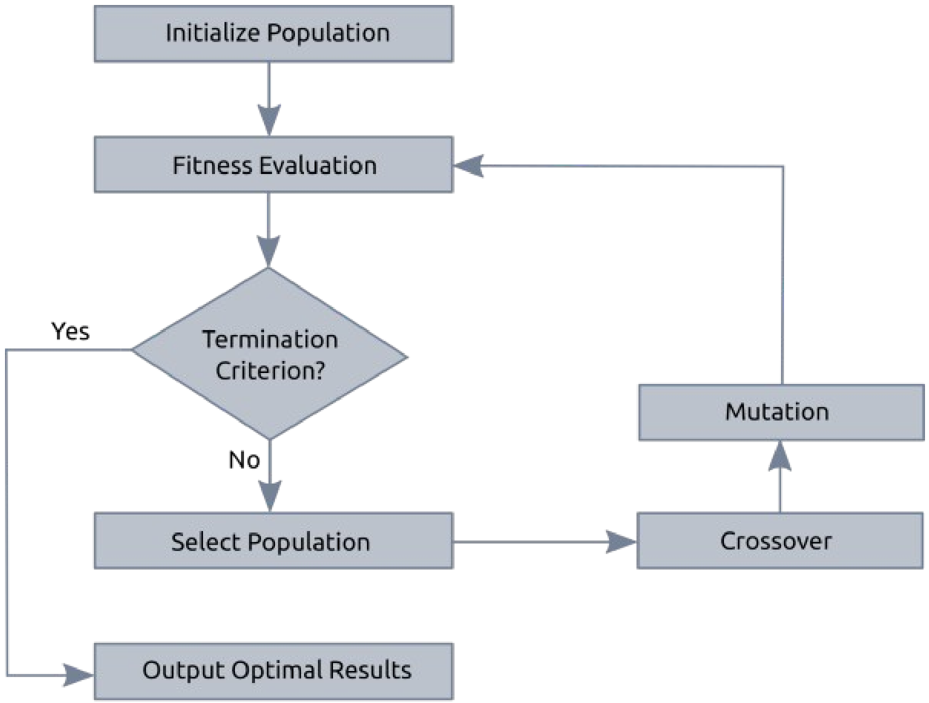
- Used in hadron synchrotrons
- Bunch-by-bunch position calculation
- Large buffers for ADC and position data storage
- Tune measurement, fft processing, slow monitoring, closed orbit feedback functionality etc.
- Accessories: Amplifier 110, SER II module, GDX module







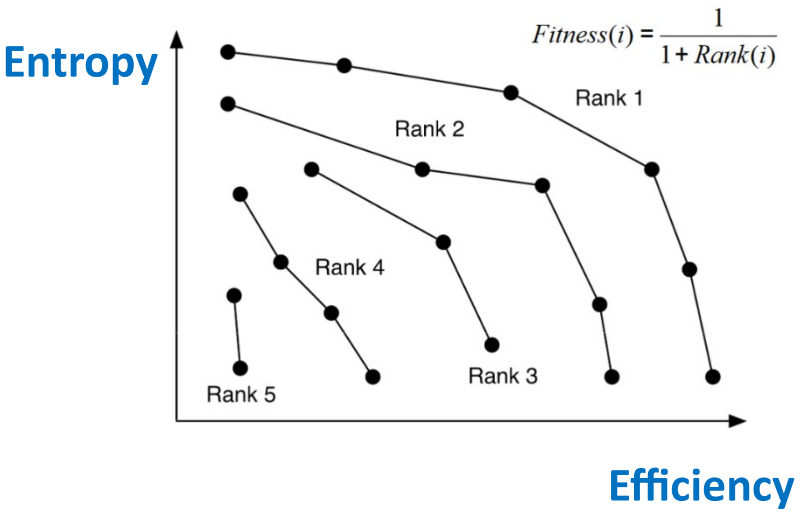
Architecture of control system for corrective magnets



Basic structure of Genetic Algorithm

Genetic algorithms (GA) is most suitable to control objects with unknown mathematical model.

The fitness function for a given candidate (called **Chromosome**) is determined by calculating the difference between the desired output and output found.



WHY NOT TO USE QUANTUM COMPUTING?

Quantum computing(QC) is a quickly growing field of research thanks to recent hardware advances. The quantum mechanical properties of quantum computers allow them to solve certain families of problems faster than classical computers.

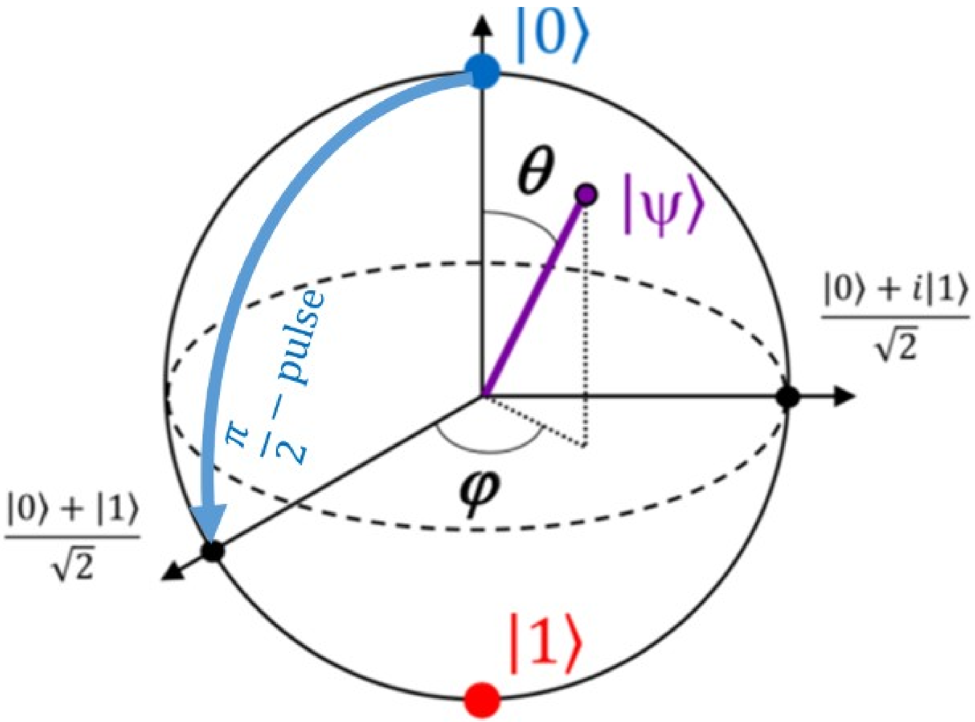
A quantum algorithm solving such a problem is Grover's algorithm, which finds an element in an unordered set faster than any classical search algorithm. So we can thus benefit from quantum computers in optimization problems, ***machine learning***, sampling of large data sets, forecasting etc.

Grover's algorithm enables one to find a specific item within a randomly ordered array of N items using $O(\sqrt{N})$ operations. By contrast, a classical computer would require $O(N)$ operations to achieve this.

The qubit (or quantum bit) is the basic container of information in a QC, replacing the bit in a conventional computer. The qubit can be in both ground and excited states at the same time. The most straightforward example is the spin. A spin qubit relies on a spin degree of freedom of either electronic or nuclear nature, which can hold a bit of quantum information for very long times. There are many other examples of qubits: two different polarizations of a photon, two energy states of an electron spinning around atom, etc. The quantum computer is fundamentally different than a classical computer due to two distinct properties of qubits. The first property is 'quantum superposition'. The second one is 'quantum entanglement'.

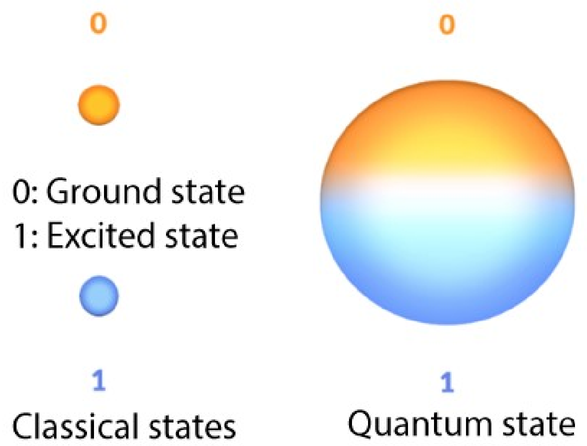
Qubit

Consider a system with two basis states, call them $|0\rangle$ and $|1\rangle$. A classical bit of data can be represented by a single atom that is in one of the two states denoted by $|0\rangle$ and $|1\rangle$.



$$|\psi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\varphi}sin\frac{\theta}{2}|1\rangle$$

The Bloch sphere provides a useful means of visualizing the state of a single qubit and operations on it. Any point on this sphere represents a linear combination of the 0 and 1 states with complex coefficients. A $\pi/2$-pulse 'rotates' a qubit from the 0-state to a superposition state.



The basic unit of information in a QC is the quantum bit (qubit) which can be in any linear combination of ground and excited sates.

Quantum Superposition

A quantum state of a qubit is in a continuous state between “0” and “1” until the qubit is measured. The outcome can only be “0” or “1”. Therefore, a qubit is a continuous object and its quantum state is given by $|\psi\rangle = \alpha 1|0\rangle + \alpha 2|1\rangle$, where α1 and α2 are complex amplitudes.

If we measure this in the computational basis, we obtain the $|0\rangle$ state with probability $|\alpha 1|^2$ or the $|1\rangle$ state with probability $|\alpha 2|^2$, where $|\alpha 1|^2 + |\alpha 2|^2 = 1$.

If one qubit can be in the superposition of two classical states, two qubits can be in a superposition of four, and ***N*** qubits can be in a superposition of 2^N: $|0\rangle, |1\rangle, |2\rangle, \ldots, |N-1\rangle$.

Therefore, a quantum register of *N* qubits is given by:
$|\psi\rangle = \alpha 0|0\rangle + \alpha 1|1\rangle + \ldots \underline{\boldsymbol{+ \alpha K|K\rangle}} + \ldots + \alpha 2N-1|2n-1\rangle$, where $\sum|\alpha|^2 = 1$.

The only thing we can say before the measurement is that we will observe state ***K*** with probability $|\alpha K|^2$.

Quantum Entanglement

The second property that distinguishes QCs from conventional computers is entanglement. If two qubits are ‘entangled’ there is a correlation between these two qubits. If one qubit is in one particular state, the other one has to be in another particular state. If two electrons become entangled, their spin states are correlated such that if one of the electrons has a spin-up, then the other one has a spin-down after measurement.

This property was pointed by Albert Einstein in 1935. The creation and manipulation of entangled states(with a help of quantum gates) plays a central role in quantum information processing.

For example, an electron in an atom can be in either the ground state or any excited state. By shining light on the atom with appropriate energy and for an appropriate length of time, it is possible to move the electron from the ground to excited state or vice versa. More interestingly, by reducing the time we shine the light, an electron initially in the state $|0\rangle$ can be moved halfway between $|0\rangle$ and $|1\rangle$.

Similarly to the previous example, the qubits in a quantum computer are first initialized, say all spin-up (zero state $|0\rangle$ on the Bloch sphere) by applying a large, static magnetic field of around one Tesla. An electromagnetic pulse is then applied to each qubit individually to bring the qubit into any possible state on the Bloch sphere. The actual qubit state depends on the pulse amplitude and time.

For instance, a superposition state can be created when using a so-called $\pi/2$-pulse (Hadamard Gate), which ‘rotates’ the qubit from the top of the Bloch sphere to the equator (***blue arrow***).

Quantum Gates

- NOT Gate is called bit flip gate (X). The output of this gate is $c1|0\rangle+c0|1\rangle$ when the input is $c0|0\rangle+c1|1\rangle$. In matrix form, this is:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

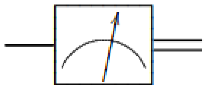
- Controlled-NOT is a two-qubit gate (CNOT). It negates the second bit of its input if the first bit is $|1\rangle$ and does nothing if the first bit is $|0\rangle$. Therefore, CNOT is a quantum gate which is different than NOT and the output depends on the first input. The first qubit is called the control qubit, the second the target qubit (**this gate is used for entanglement**):

$$CNOT = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- Hadamard Gate (H) gate is used to create the quantum superposition (π/2-pulse), specified by $H|0\rangle=1\sqrt{2}|0\rangle+1\sqrt{2}|1\rangle$ when starting from the zero state. The Hadamard gate outputs $|0\rangle$ or $|1\rangle$ with equal probability:

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

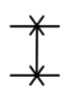
There are other quantum gates in use, such as the phaseflip, phaseshift, and Toffoli gate among others*. A measurement gate is used to project a qubit's state onto the basis vectors $|0\rangle$ and $|1\rangle$. This step is necessary for extracting a result from the quantum computation and is the only non-reversible quantum gate. Once a qubit is measured its quantum state is destroyed.

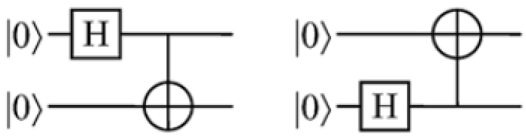
$|q_0\rangle$ 

* http://www.libquantum.de/api/1.1/Basic-quantum-operations.html

Quantum gates symbols

Frequently used unary and binary quantum gates

Gate	Unitary matrix	Symbol
NOT	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\boxed{\neg}$
Hadamard	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\boxed{\text{H}}$
Square NOT	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$	$\boxed{\sqrt{\neg}}$
Controlled NOT	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	
Simple rotation	$\begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$	$\boxed{\text{U}(\theta)}$
Swap	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	



Quantum circuits for the production of 2 qubit entangled states.

Up to this point, our discussions on the quantum computing model have been quite abstract. The purpose of this discussion was to explicitly provide important initial information. **Now let's see how everything told at this point looks in practice.** Let's look at existing tools for quantum computing.

There is a **libquantum*** - quantum computing emulation library and Grover's algorithm implemented by its means. A test implementation of this algorithm is included in the library distribution.

Based on the principles of quantum mechanics the libquantum library is a freeware library of quantum computing emulation written in C. It supports the data type "quantum register", the operation of applying quantum gates to the quantum register and measurement operations, but it works too slow (**hovever this can be implemented using CUDA, which can increase the speed of calculations by several orders of magnitude**).

Let's look how to implement Grover's algorithm using real quantum computer.
let's find an element with an optimal fitness value in the chromosome indexed array:

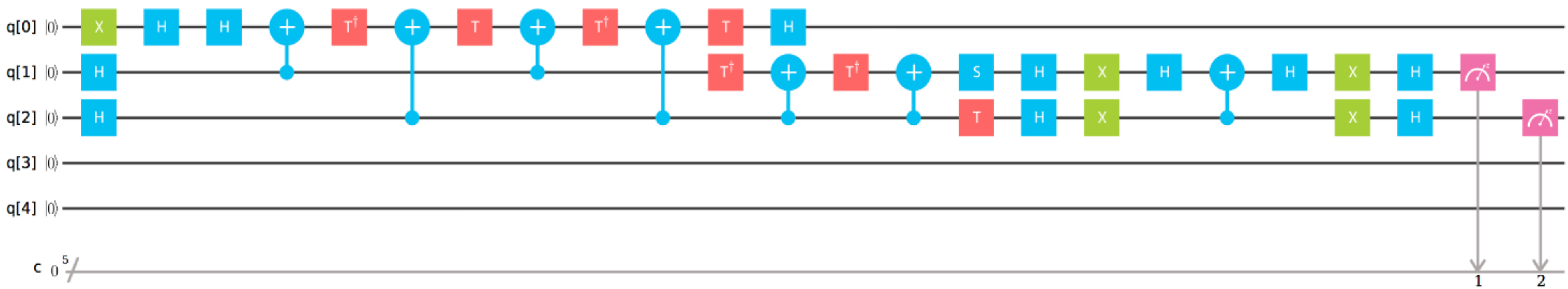
Fitness_values {3, 2, 0, 1}

chromosome_index {0, 1, 2, 3}

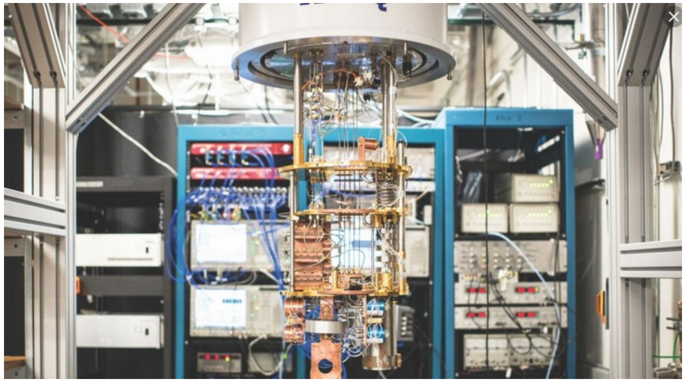
Let’s assume that 2 is desired fitness value.
It’s position in **chromosome_index** list corresponds to 1.
Then the task of the Grover algorithm will be to find the index of the desired fitness value 2.

*http://www.libquantum.de

A quantum circuit is a superposition of quantum gates, and applying it to a quantum register is equivalent to a sequential, in the correct order, application of individual gates. Qubit quantum state $|\psi\rangle$ rotations on the Bloch sphere are implemented by quantum gates application. These quantum register state rotations increases the probability of state which represents desired value, given by «Oracle» (a special quantum gate combination).



The circuit that was executed on IBM's 5-qubit quantum computer (ibmqx4 was used). The first two time slots correspond to the state preparation. The next 13 time slots implement a Toffoli gate. The next 7 time slots implement the $2|\psi\rangle\langle\psi| - I$ operator, and the final two time slots are used for observing x_1 and x_2.

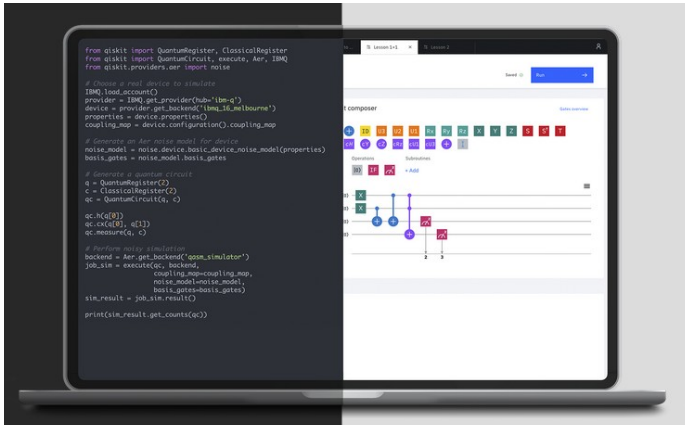


IBM have several quantum computers available to the public through their cloud service IBMQ. The ibmqx5 is a superconductivity-based 16-qubit quantum computer available through a Python-based programming interface called QISKit.

IBM Q Account is the doorway for accessing the full suite of cloud-based IBM Q quantum systems and simulators.

```
from qiskit import IBMQ
from qiskit.providers.ibmq import least_busy

# Load local account information
provider = IBMQ.load_account()

# Get the least busy real quantum system
backend = least_busy(provider.backends(simulator=False))
print(backend, backend.status().pending_jobs)
```

Qiskit environment allows to send jobs to IBM Q systems. Qiskit is an open-source framework for quantum computing. The primary version of Qiskit uses the Python programming language. The QISKit interface provides a built in mapper for mapping a qubit in the code to a hardware qubit.



4-qubit Grover's Algorithm

```
from qiskit import QuantumProgram
import math

import Qconfig
from IBMQuantumExperience import IBMQuantumExperience
api = IBMQuantumExperience(Qconfig.APItoken,
    {'url':Qconfig.config["url"]})
from qiskit.backends import discover_remote_backends
remote_backends = discover_remote_backends(api)

qp = QuantumProgram()

pi = math.pi

qr = qp.create_quantum_register('qr', 4)
cr = qp.create_classical_register('cr', 4)
qc = qp.create_circuit('Grover', [qr], [cr])
shots = 8192
```

```
######################
######## init ########
######################
qc.h(qr[0])
qc.h(qr[1])
qc.h(qr[2])
qc.h(qr[3])

######################
### Oracle for 0010 ###
######################
qc.x(qr[0])
qc.x(qr[2])
qc.x(qr[3])

qc.cu1(pi/4, qr[0], qr[3])
qc.cx(qr[0], qr[1])
qc.cu1(-pi/4, qr[1], qr[3])
```

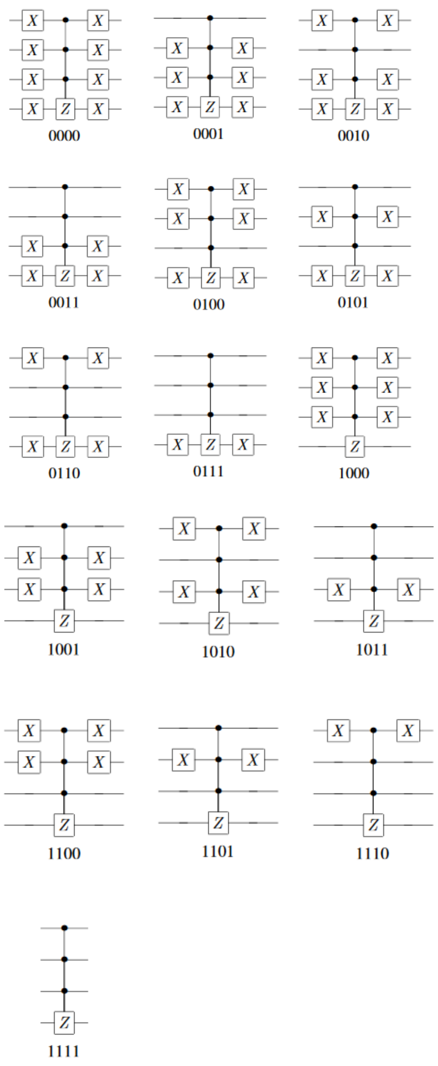
```
qc.cx(qr[0], qr[1])
qc.cu1(pi/4, qr[1], qr[3])
qc.cx(qr[1], qr[2])
qc.cu1(-pi/4, qr[2], qr[3])
qc.cx(qr[0], qr[2])
qc.cu1(pi/4, qr[2], qr[3])
qc.cx(qr[1], qr[2])
qc.cu1(-pi/4, qr[2], qr[3])
qc.cx(qr[0], qr[2])
qc.cu1(pi/4, qr[2], qr[3])

qc.x(qr[0])
qc.x(qr[2])
qc.x(qr[3])

######################
#### Amplification ####
######################
qc.h(qr[0])
qc.h(qr[1])
qc.h(qr[2])
qc.h(qr[3])

qc.x(qr[0])
qc.x(qr[1])
qc.x(qr[2])
qc.x(qr[3])
```

```
######## cccZ #########
qc.cu1(pi/4, qr[0], qr[3])
qc.cx(qr[0], qr[1])
qc.cu1(-pi/4, qr[1], qr[3])
qc.cx(qr[0], qr[1])
qc.cu1(pi/4, qr[1], qr[3])
qc.cx(qr[1], qr[2])
qc.cu1(-pi/4, qr[2], qr[3])
qc.cx(qr[0], qr[2])
qc.cu1(pi/4, qr[2], qr[3])
qc.cx(qr[1], qr[2])
qc.cu1(-pi/4, qr[2], qr[3])
qc.cx(qr[0], qr[2])
qc.cu1(pi/4, qr[2], qr[3])
####### end cccZ #######

qc.x(qr[0])
qc.x(qr[1])
qc.x(qr[2])
qc.x(qr[3])

qc.h(qr[0])
qc.h(qr[1])
qc.h(qr[2])
qc.h(qr[3])
```

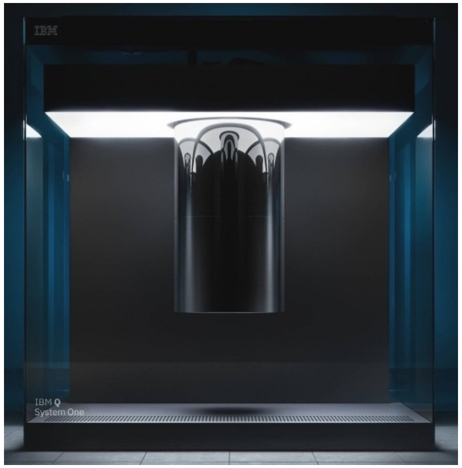
```
########################
####### Measure ########
########################
qc.barrier(qr)
qc.measure(qr[0], cr[0])
qc.measure(qr[1], cr[1])
qc.measure(qr[2], cr[2])
qc.measure(qr[3], cr[3])

# submit job #
qp.execute(['Grover'], backend='ibmqx5', shots = shots,
    max_credits = 5, timeout=1)
```

«Oracle» is a special quantum gate combination which corresponds to desired(sought) value.

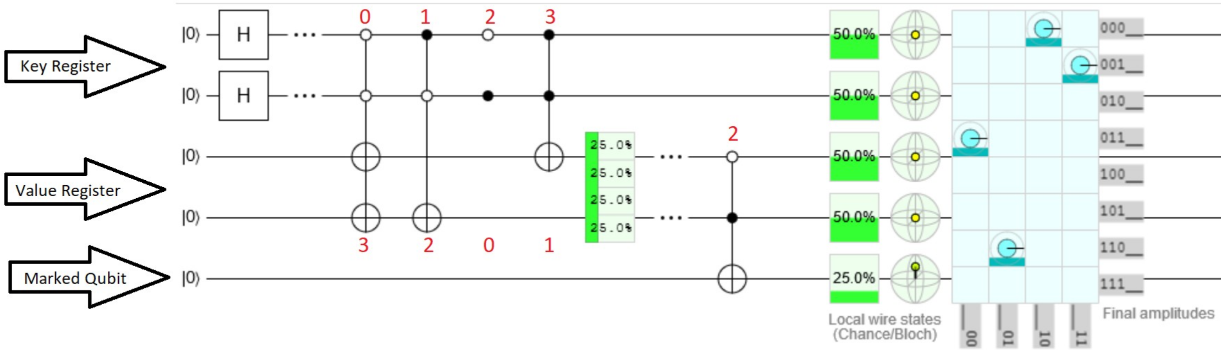
Oracles

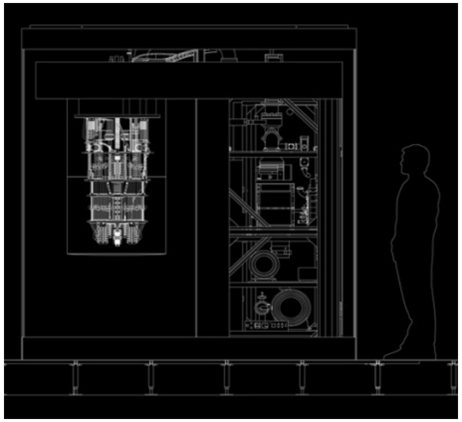


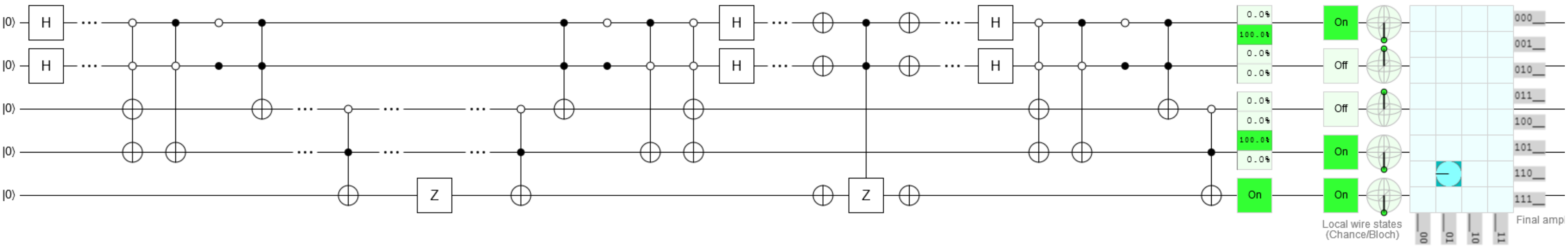
To search our indexed array we have to introduce a special combination of quantum entanglement gates, which «connects» values and indexed (keys).



IBM Q System One has 20 qubits — first commercial quantum computer (introduced by IBM in January 2019).







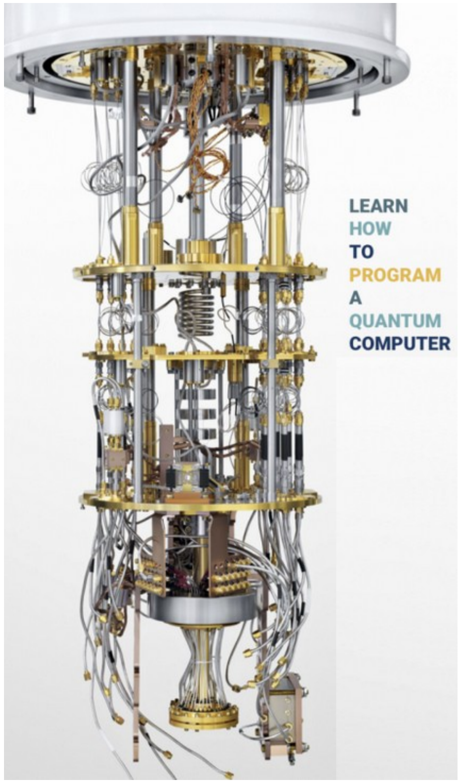
Conclusion

Quantum computers work according to principles defined by quantum mechanics and the theoretical basis is now well defined. As we have seen, the power of quantum algorithms seems to be derived from the properties of entangled states and amount of qubits. It is thus very important to keep track of the progress of quantum computing technologies.

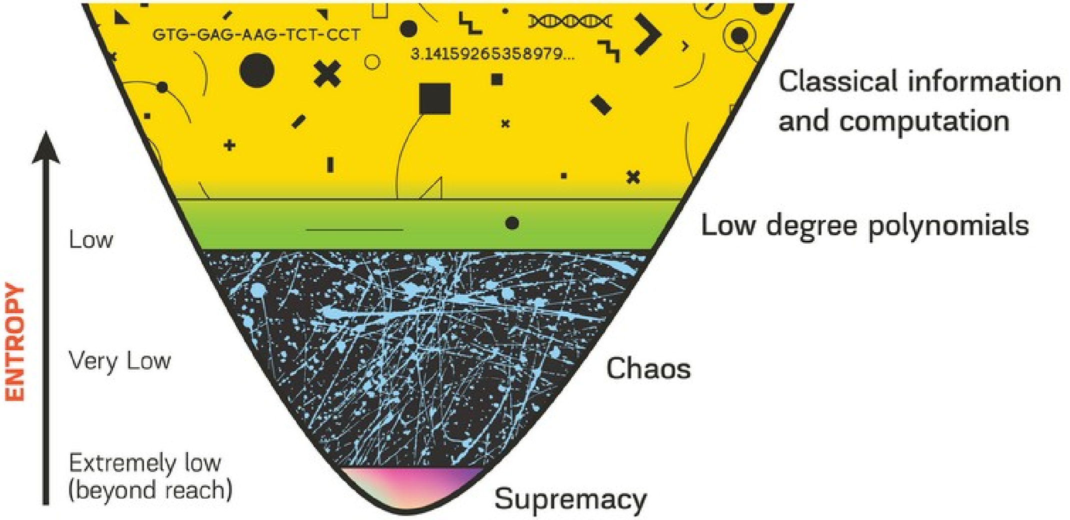
IBM prepares 53-Qubit Quantum Computer for Launch in October 2019
2^{53} = 9 007 199 254 740 992

It will allow to implement Grover's search on a real indexed array of 2^{26} = 67 108 864 values.

A quantum computer with 100 qubits can search an indexed array of 2^{49} = 562 949 953 421 312


LEARN
HOW
TO
PROGRAM
A
QUANTUM
COMPUTER

Thank you


GTG-GAG-AAG-TCT-CCT
3.14159265358979...
Classical information and computation
Low degree polynomials
Chaos
Supremacy
ENTROPY
Low
Very Low
Extremely low (beyond reach)