Welcome to the Quantum Computer Science Proposers' Day

10:00 – 10:05	Introductory Remarks	Dr. Mark Heiligman QCS Program Manager
10:05 – 10:30	IARPA Overview	Dr. Lisa Porter IARPA Director
10:30 – 11:45	QCS Overview	Dr. Mark Heiligman
11:45 – 12:00	Contracting Overview	Ms. Diane Hodor Contracting Officer
12:00 – 1:00	Lunch Break – On your own	
1:00 – 2:30	Proposers' Presentations	
2:30 – 2:45	Administrative Remarks (Gov't Representatives Depart)	Dr. Mark Heiligman
2:45 – 3:00	Break	
3:00 – 5:00	Posters and Teaming Discussions	

QCS Proposers' Day
IARPA-BAA-10-02 Overview

Dr. Mark Heiligman
Intelligence Advanced Research Projects Activity (IARPA)
Office of the Director of National Intelligence (ODNI)

Disclaimer

- This presentation is provided solely for information and planning purposes.
- The Proposers' Day Conference does not constitute a formal solicitation for proposals or proposal abstracts.
- Nothing said at Proposers' Day changes the requirements set forth in a BAA.
- Any conflict between what is said at Proposers' Day and what is in a BAA will be resolved in favor of the BAA.

Goals of Proposers' Day

- Familiarize participants with IARPA's interest in Quantum Computer Science – Please provide feedback, this is your chance to alter the course of events.

- Foster discussion of synergistic capabilities among potential program participants, AKA teaming. Take a chance, someone might have a missing piece of your puzzle.

Schedule

- **Full Proposals are due ~45 days after BAA is published.**

Once the BAA is released, questions can only be answered in writing on the program website.

Agenda

- Program Overview
- Program Metrics and Milestones
- Award Information
- Eligibility Information
- Proposal Review Information
- Q&A session (please write your questions on the provided cards)

QCS Program Proposers' Day

Program Overview

Why is Quantum Computing of Interest?

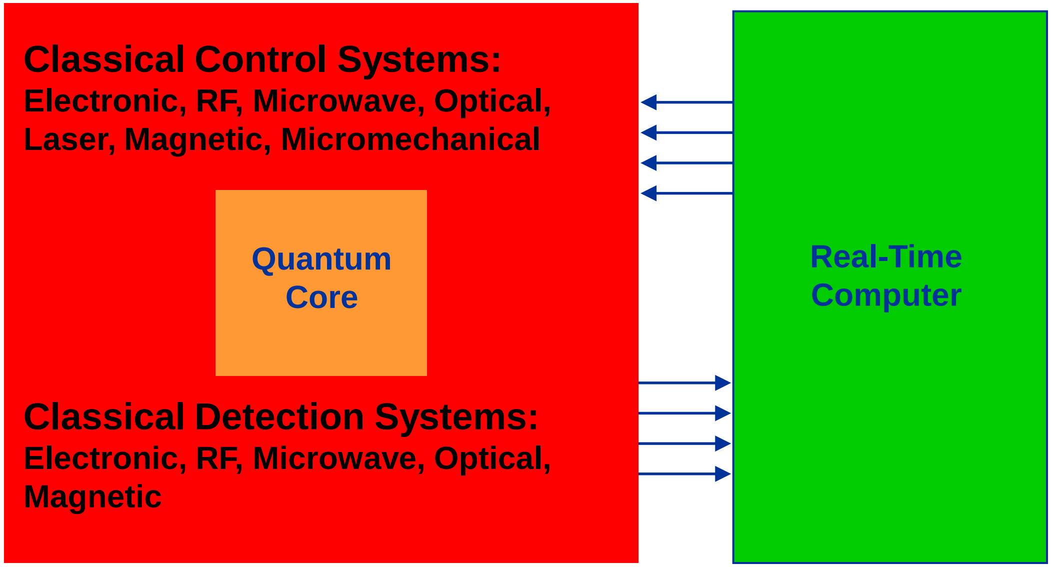
Algorithm Name	*Classical Complexity*	*Quantum Complexity*
Elliptic Curve Discrete-Log (Shor's Algorithm)	$O(2^{N/2})$	$O(N^3)$
Dihedral Hidden Subgroup	$O(2^N)$	$O(2^{\sqrt{N}})$
Hash Collision Finding	$O(N^{1/2})$	$O(N^{1/3} \log(N))$
Element Distinctness	$O(N \log(N))$	$O(N^{2/3} \log(N))$
NAND tree	$O(2^{.753N})$	$O(2^{.5N})$
Quantum Search (Grover's Algorithm)	$O(N)$	$O(N^{1/2})$

Quantum computing appears to offer better algorithms than classical computing for some problems

However, complexity analysis alone does not provide realistic estimates of execution times.

How is a quantum algorithm implemented?

- **Executing a quantum algorithm means performing a massive computer controlled quantum physics experiment.**
- **The full specification of how this experiment is to be carried out constitutes the <u>implementation</u> of a quantum algorithm.**



The quantum system core is expensive and fragile; therefore, we must minimize its number of qubits and operations.

Quantum Computing: Challenges and Approaches

Quantum computing faces significant challenges:

- Fault-tolerant quantum computation requires prohibitive resource overhead (additional qubits and operations) and improved quantum control.
- There is a lack of tools for accurately estimating resource overhead, performance of fault-tolerant strategies, and for investigating potential system trade-offs.

The QCS program will meet these challenges by:

- Improving quantum control protocols to achieve input error-rate thresholds required by quantum error correction protocols.
- Improving and developing new quantum error correction protocols to greatly reduce their resource overhead and improve performance under various error models.
- Creating quantum algorithm implementation tools that produce optimized and credible resource and performance estimates.

The QCS Program's overarching goal is to accurately estimate and significantly reduce the computational resources required to implement quantum algorithms on a realistic quantum computer.

Areas NOT included in this program

- Physical experiments/theoretical development relating to specific qubit technologies
- Full simulations of "small" quantum computer systems
- Discovery of new quantum algorithms
- Structure of complexity classes
- Optimization of specific algorithms (although general optimizing compilers are expected)
- Computational paradigms other than the quantum circuit model

QCS Program Proposers' Day

Program Activities

QCS: Deliverables

1) **Capability to compute resource requirements for implementing quantum algorithms given:**
   - **Quantum program and associated parameters**
   - **Error correction protocol and control protocol**
   - **Physical Machine Description**
     - **Provides technology-dependent specification of native hardware needed to translate a quantum program to machine code.**

   Initial baseline estimates may be manual, but a software toolbox shall be developed to automate the process

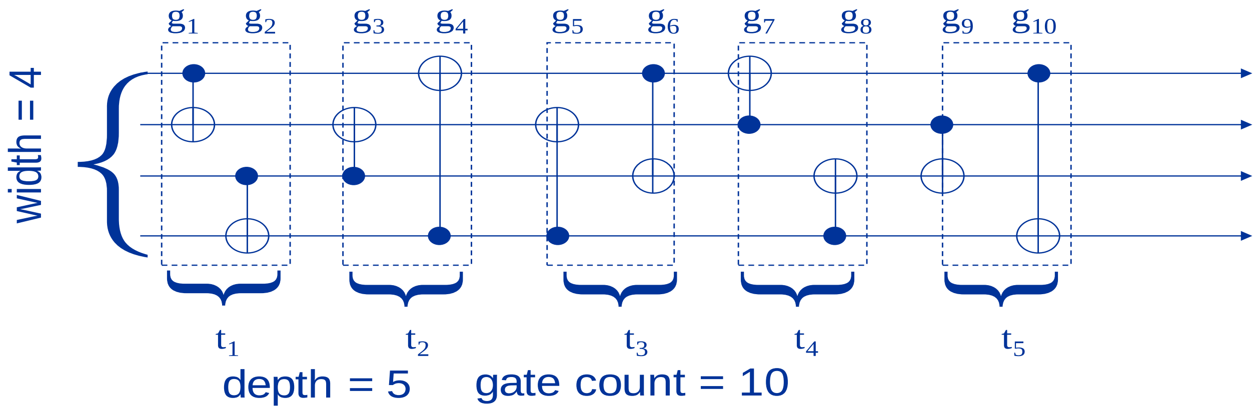
2) **Innovative optimization techniques, quantum error correction protocols, and quantum control protocols, with supporting software, that will be used to reduce resource requirements and improve fault-tolerant performance.**

Software deliverables are research tools not products.

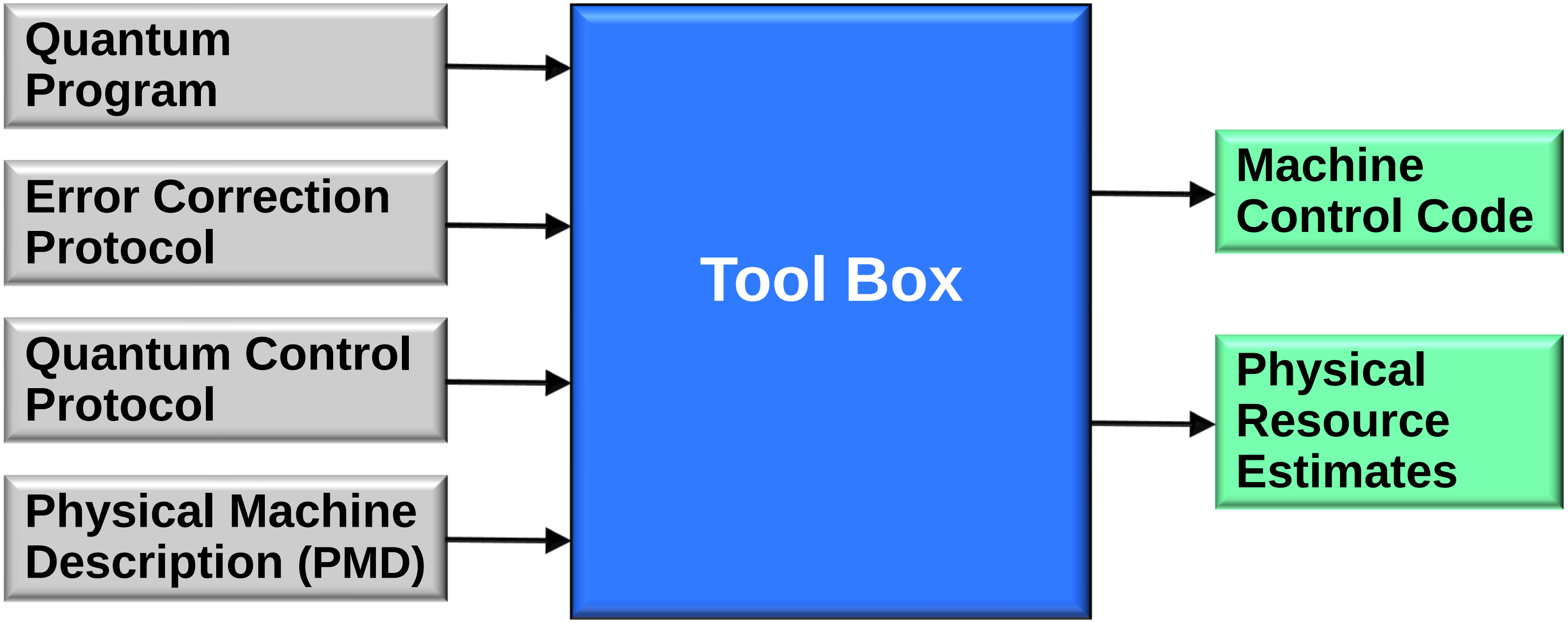
QCS: Resource Estimates

- In the QCS program, we are only interested in the circuit model of quantum computation .

- Resource estimates will, therefore, be provided in terms of circuit resources, including width (number of qubits), depth (number of "time-slices"), gate operation count, and will include all quantum control and quantum error correction overhead.



QCS: Software Toolbox – Resource Estimation

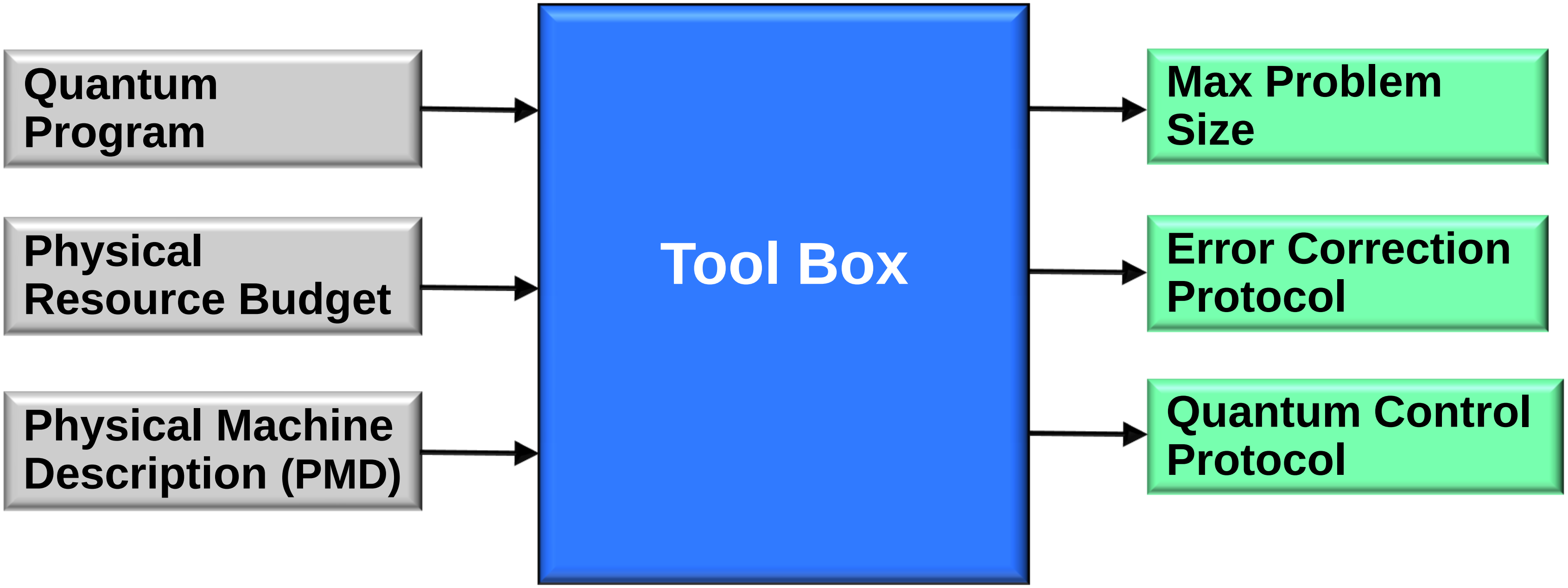


In one typical application of the toolbox, a user will:

- Compose a quantum program
- Specify the quantum error correction protocol
- Specify the quantum control protocol
- Provide the physical machine description

User receives the PMD-based machine control code and resource estimates

QCS: Software Toolbox
- Inverse Problem



In a typical inverse-problem application of the toolbox, a user will:

- Compose a quantum program
- Specify the quantity of available physical resources
- Provide the physical machine description

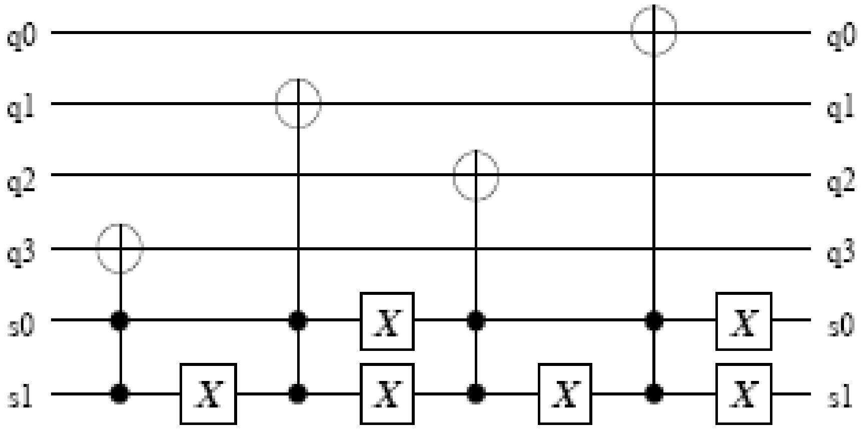
User receives maximum problem size solvable with given resources, along with corresponding error correction and control protocols

Example: High-level Quantum Programming

Example: Subroutine expressed in a high level language

```
cond qufunct demux(quconst s,qureg q) {
  int i;
  int n = 0;
  for i=0 to #s-1 {                    // accumulate content of
    if s[i] { n=n+2^i; }               //   selection register in a
  }                                    //   classical variable
  Not(q[n]);                           // flip selected output qubit
}
```

A graphical depiction of corresponding circuit model representation of subroutine for #s=2



Source: Ömer, "Classical Concepts in Quantum Programming," quant-ph/0211100v2

QCS is looking for innovative high-level programming techniques for expressing and optimizing quantum programs.

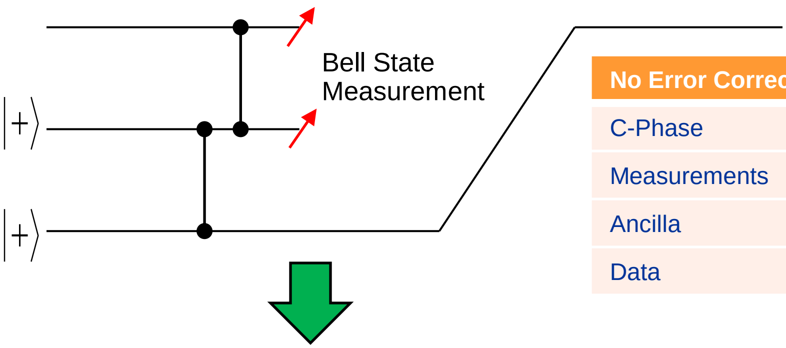
Quantum Error Correction and Quantum Control

- ❑ **Reliable computers can be constructed from faulty components when the components' error rate falls beneath a certain threshold**
  - ➢ **Classical Theorem (von Neumann, 1956)**
  - ➢ **Quantum Version (Shor et al., 1999)**

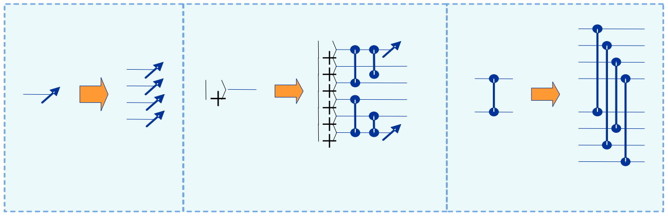
- ❑ **Quantum fault-tolerance relies on quantum error correction and quantum control protocols, at the expense of additional physical resources.**

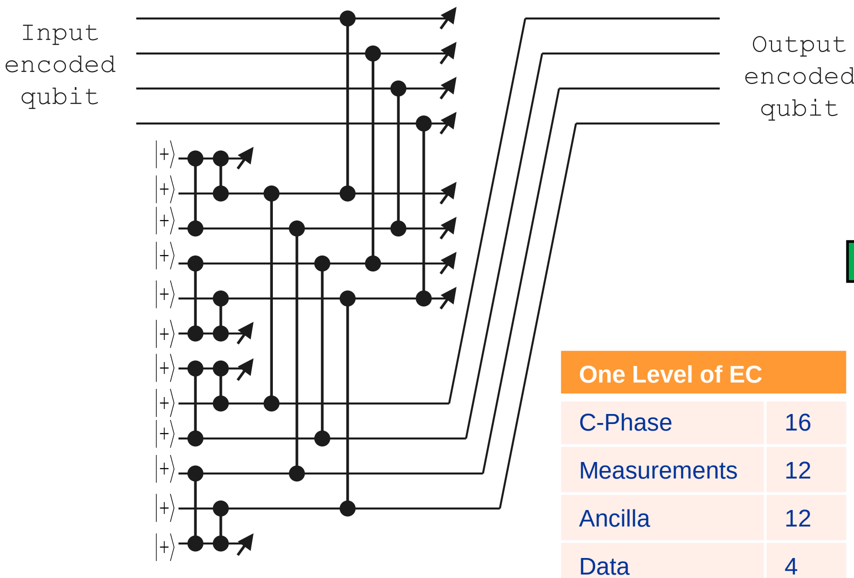
The QCS program seeks reliable quantum computation with substantial reductions in overhead costs via innovative quantum error correction and quantum control protocols.

Example: Quantum Error Correction

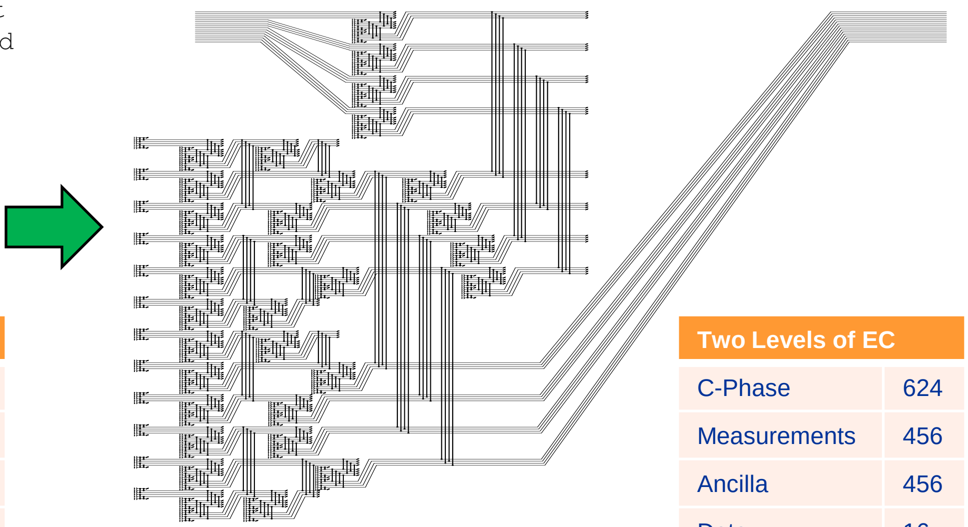


No Error Correction	
C-Phase	2
Measurements	2
Ancilla	2
Data	1





One Level of EC	
C-Phase	16
Measurements	12
Ancilla	12
Data	4



Two Levels of EC	
C-Phase	624
Measurements	456
Ancilla	456
Data	16

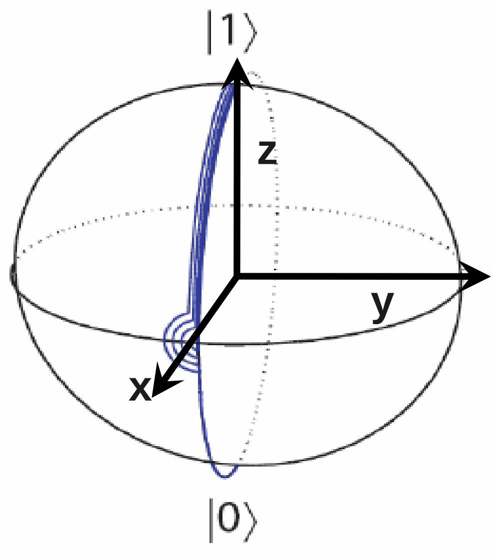
The QCS Program is looking for ways to dramatically reduce quantum error correction resource overhead while maintaining system reliability.

***From Henry Haselgrove**

Example: Quantum Control

Simple Example: NOT-Gate

- A π rotation about the y-axis is accomplished with three rotations rather than one:
  - $Rot(\pi,y) = Rot(\pi/2,y) \circ Rot(\pi,x) \circ Rot(\pi/2,y)$
  - This simple control protocol improves gate fidelity for rotation around y axis: $O(\varepsilon_y)$ errors → $O(\varepsilon_y^2)$ errors



Although improved quantum control incurs additional resource overhead, it has the potential to achieve required quantum error correction input error-rate thresholds.

QCS: The Case for Correctness

- **The toolbox generates machine code and resource estimates.**

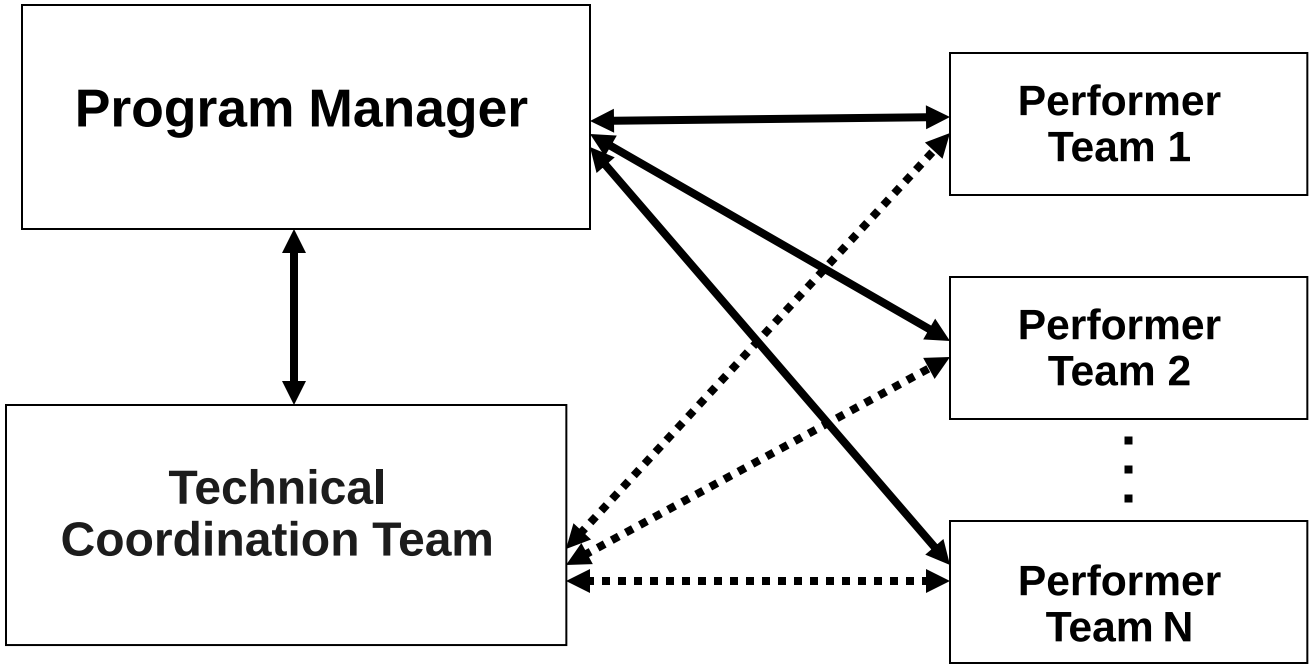
- **A quantum computer capable of executing the machine code does not exist.**

- **Methodology and tools (e.g., mathematical analyses) are needed to support strong statements about the correctness of the machine code and resource estimates.**

QCS Program Proposers' Day

Additional QCS Program Information

QCS: Program Management Structure



A Technical Coordination Team comprised of personnel from organizations that are ineligible to compete in the QCS Program will provide test and evaluation capabilities.

QCS: Government Furnished Information

- The government will provide the following information:
  - List of quantum algorithms, and associated problem sizes
  - Baseline quantum error correction protocols
  - Baseline quantum control protocols
  - List of the resources that are to be estimated
  - Baseline Physical Machine Descriptions (PMDs)
    - Multiple PMDs will enable the QCS Program to encompass a wide range of qubit technologies within the circuit model of quantum computing.
    - A physical machine description format will be provided.
  - Machine Control Language(s) (MCL)
    - The MCL encodes simple, technology-independent control instructions that translate into technology-dependent commands.

These items will be provided at program kick-off and refined throughout the program in an iterative process with input from performers.

QCS Program Proposers' Day

Program Phases, Milestones, Metrics

QCS Program: Targets

The QCS Program's overarching goal is to accurately estimate and significantly reduce the computational resources required to implement quantum algorithms on a realistic quantum computer.

- **Reduce resource requirements for credible implementations of quantum algorithms on realistic quantum computers by at least four orders of magnitude, through combining:**
  - **Reductions from innovative quantum error correction.**
  - **Reductions from innovative quantum control.**
  - **Reductions from general optimization techniques.**

- **Improve protocol performance in**
  - **Fault-tolerant threshold (keeping error correction overhead the same).**
  - **Controlled gate fidelity (keeping bandwidth the same).**

- **QCS will begin by establishing the baseline ...**

QCS: Activity by Phase

Phase	Activities	Milestones & Metrics
1 **"Baseline"** **(1 year)**	Estimate baseline resources for GFI algorithms set, with GFI quantum error correction, control protocol and physical machine description. Create detailed plans for: (a) toolbox implementation; (b) toolbox validation methods; (c) development and use of innovative optimizations, quantum error correction and control protocols via the toolbox.	Baseline resource estimates established; Viable toolbox plan; Credible quantum error correction, control protocol and optimizations plan.
2 **(2 years)**	Complete toolbox-v1; capable of full operation; replicate phase 1 results; show ability to modify quantum error correction, control protocols , and physical machine descriptions; Establish methodology for establishing validity of machine code and estimates on full GFI algorithm set; Update plans with the goal of a viable path to a further 100x resource reduction.	Reduce baseline resource estimates by x100; Produce convincing validation of toolbox functionality and program correctness. Deliver quantum error correction and control protocols used to achieve resource reductions.
3 **(2 years)**	Complete toolbox-v2; replicate phase 2 results; Demonstrate ability to handle wide variety of quantum error correction, control protocols , and physical machine descriptions; Perform toolbox-based tradeoff studies on specific quantum problems (and sizes); Solve examples of inverse problem.	Reduce baseline resource estimates by x10,000; improve fault-tolerant threshold by 5x. Deliver quantum error correction and control protocols used to achieve resource reductions.

QCS PROGRAM PROPOSERS' DAY

AWARD INFORMATION

QCS: Award Plan

- 5-year effort starting FY2010
  - Phase 1 – Base Period – 1 year
  - Phase 2 – Option Period – 2 years
  - Phase 3 – Option Period – 2 years

- Criteria for moving to Phase 2: sufficient progress in achieving Phase 1 goals

- Criteria for moving to Phase 3: sufficient progress in achieving Phase 2 goals

- Multiple awards anticipated, depending upon
  - quality of the proposals received
  - availability of funds

QCS PROGRAM PROPOSERS' DAY

ELIGIBILITY INFORMATION

Eligibility Information

- **Collaborative efforts/teaming strongly encouraged**
  - **Content, communications, networking, and team formation - responsibility of proposers**
- **Non-US organizations and individuals may be able to participate**
  - **Must comply with Non-Disclosure Agreements, Security Regulations, Export Control Laws, etc, as appropriate**
  - **Specific guidance for non-US participation will be provided in the BAA**
- **Other Government Agencies, Federally Funded Research and Development Centers (FFRDCs), University Affiliated Research Centers (UARCs), and any organizations that have a special relationship with the Government, including access to privileged and/or proprietary information, or access to Government equipment or real property, are not eligible to submit proposals under this BAA or participate as team members under proposals submitted by eligible entities.**

QCS PROGRAM PROPOSERS' DAY

PROPOSAL REVIEW INFORMATION

Proposal Guidance

- Your proposal should include a full discussion of the technical approach that will be used to meet the program goals.
- Programmatic issues that should be addressed in the proposal:
  - Your team's current technical capabilities
  - Key resources needed (not currently available to your team), to include capital equipment and special expertise (teaming will likely play an essential role in providing special expertise). The risk in acquiring these key resources, and mitigation strategies, should be indicated as well.
  - A teaming plan along with the roles and responsibilities of each member of the research team
  - End of phase and some immediate milestones are set, but it is expected that other intermediate milestones that are on the critical path of the proposed approach will be offered.
  - A schedule of all milestones including a clearly charted description of the various risk mitigation strategies that will be undertaken to achieve program goals

Evaluation Criteria

- **Evaluation criteria in descending order of importance are:**
  - **Overall Scientific and Technical Merit**
  - **Effectiveness of Proposed Work Plan**
  - **Relevance to IARPA Mission and QCS Program Goals**
  - **Relevant Experience and Expertise**
  - **Cost Realism**

- **All proposals will be evaluated by a board of qualified government reviewers. Each proposal will be evaluated by at least three reviewers.**

Teaming

- Because of the many challenges presented by this program, both depth and diversity will be beneficial for overcoming these challenges.
  - Throughput – Consider all that you will need to do, all the ideas you will need to test.
    - Make sure you have enough people and expertise to do the job
    - Sufficient resources to follow critical path while still exploring alternatives – risk mitigation
  - Completeness – teams should not lack any capability necessary for success, e.g. should not rely upon results from the community at large, or some enabling technology to be developed elsewhere.
  - Tightly knit teams
    - Clear, strong, management, single point of contact
    - No loose confederations
    - Each team member should be contributing significantly to the program goals. Explain why each member is important, i.e. if you didn't have them, what wouldn't get done?
    - No teaming for teaming's sake.
- Remember, you may be very accomplished, but can you do it all?

Needed Expertise

- A successful proposal will likely show expertise in the following areas:
  - Compilers
  - High-level language development
  - Automated design tools
  - Error correcting codes
  - Control theory
  - Algorithms
  - Quantum physics
  - Software development

Point of Contact

Dr. Mark Heiligman

Program Manager

IARPA, Incisive Analysis Office

Office of the Director of National Intelligence
Intelligence Advanced Research Projects Activity
Washington, DC 20511

Phone: 301-226-9100

Fax: 301-226-9137

Electronic mail dni-iarpa-baa-10-02@ugov.gov

(include IARPA-BAA-10-02 in the Subject Line)

Program website: www.iarpa.gov

QCS Program Proposers’ Day

Thank you!

Any questions?

Watch for the QCS BAA in the near future at

http://www.iarpa.gov/solicitations.html