Adventures in Quantum Computing

A New Frontier for Design-Space Exploration and Performance Modeling

Jeffrey S. Vetter ·vetter@computer.org·vetter.github.io
Oak Ridge National Laboratory · MODSIM Workshop · August 2026

Executive summary

01 Fault-tolerant quantum computing is now an **architecture problem** — and DOE has set a 2028 deadline for scientifically relevant systems.

02 These machines are mostly classical systems: real-time decoding, control, and HPC integration dominate the design.

03 Predicting their physical cost is **ad hoc today** — incompatible tools, hidden assumptions, no energy models at all.

04 The MODSIM toolbox — interchange formats, cost models, design-space exploration, multi-objective optimization — **transfers directly**. We'll show our first results.

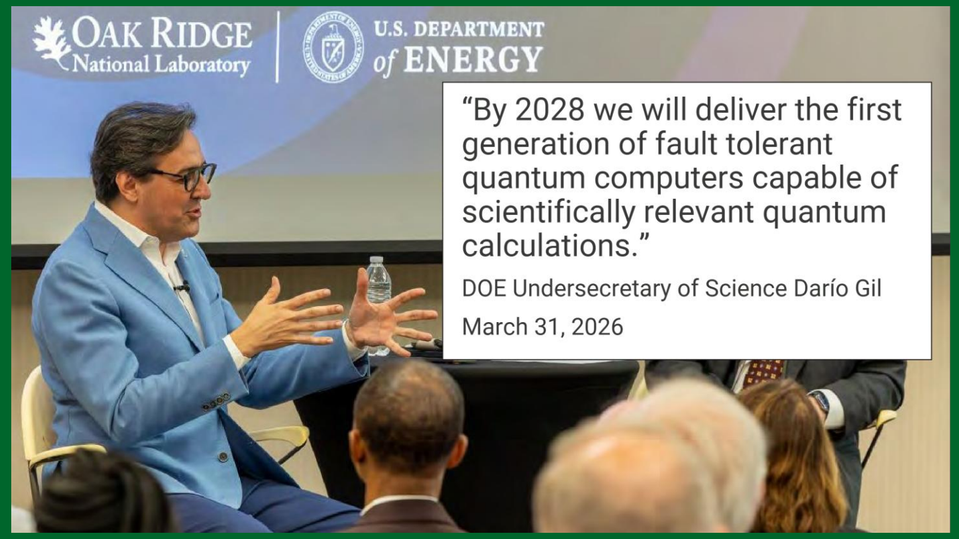
DOE QUANTUM GENESIS INITIATIVE — JUNE 2026

DOE has set a 2028 deadline for fault-tolerant quantum systems

DOE Q Competition — demonstrate fault-tolerant systems with logical qubits numbering in the low hundreds, aimed at chemistry, materials, plasma, and high-energy physics.

National Quantum Supercomputing User Facility — multiple modalities, integrated with DOE's exascale HPC, AI, and ESnet into one discovery ecosystem.

DARPA Quantum Benchmarking Initiative — in parallel: staged, independent verification of whether any architecture reaches utility scale by 2033. Eleven companies in Stage B; hardware testing to follow in Stage C.



DOE Undersecretary of Science Darío Gil announces the 2028 goal, March 2026.

Integrating quantum into the HPC center: applications

RIKEN + IBM: closed-loop quantum-centric chemistry

sample-based quantum diagonalization (SQD) iterating between ibm_kobe and Fugaku in production

Electronic structure of strongly correlated systems

iron–sulfur clusters, transition-metal catalysts — QPU samples, HPC diagonalizes

Materials, plasma, and HEP simulation

the DOE Q Competition's target science areas — all hybrid by construction

The pattern is stable across all of them: **the QPU is a sampler in an HPC-orchestrated loop** — classical pre-processing, quantum execution, classical post-processing, repeat.

Fault tolerance deepens the loop — decoding, magic-state distillation, and compilation join the classical side of the workflow.

Sizing that loop end-to-end is exactly the estimation problem this talk is about.

Template	Parameters	Scaling
Shor	n (bits)	$3n+\lceil\sqrt{n}\rceil$ qubits, $0.3n^3$ Toffoli [16]
QPE	qubits, precision	2^b controlled-U
Ham. sim.	qubits, terms, time	Trotter / QSP / qubitization
Chemistry	orbitals, electrons	DF [21] / THC / sparse
Grover	search bits	$\frac{\pi}{4}\sqrt{N/M}$ iterations

Integrating quantum into the HPC center: operations

RIKEN R-CCS · KOBE

Fugaku coupled with IBM Quantum System Two and Quantinuum's Reimei on one campus — plus the new ROQUO GPU system as the hybrid platform's classical engine.

JHPC-Quantum: two qubit modalities behind one system-software stack, operated by the HPC center itself.

LRZ / TU MUNICH · GARCHING

Euro-Q-Exa: IQM superconducting QPUs installed inside the center and coupled to SuperMUC-NG for hybrid workflows.

Munich Quantum Software Stack: the QPU as a schedulable device in the HPC resource manager, not a cloud endpoint.

ORNL · OAK RIDGE

Pathfinder, a 20-qubit IQM Radiance — IQM's first U.S. on-premises system — connected to NCCS test-bed HPC, alongside Quantum Brilliance's Quoll at OLCF.

Proving ground for an open, hardware-agnostic quantum-HPC software stack.

OUR PLAN

Scale this playbook to fault tolerance: on-premises QPUs as **first-class scheduled resources** — co-scheduling with GPU partitions for decoding, facility integration (cryogenics, power, networking), and availability engineering across quantum and classical maintenance windows.

Quantum computing in one slide

STATE

A register of n qubits holds a vector of **2^n complex amplitudes** — not 2^n parallel answers.

Yes, complex numbers — the physicists write i where you'd write j. First interoperability problem of the field.

50 qubits ≈ more amplitudes than any classical memory can store.

COMPUTE

Gates rotate that vector; a program choreographs **interference** so wrong answers cancel and right ones reinforce.

Circuits look like reversible linear algebra, not branching code.

MEASURE

Reading out collapses the state to **n ordinary bits** — one sample from the amplitude distribution.

The art is making that one sample worth exponential classical work.

The payoff is real but narrow: **factoring, quantum chemistry, materials, and simulation of quantum systems** — problems with structure that interference can exploit. It is an accelerator for select workloads, not a faster general-purpose computer.

Quantum computing is leaving the NISQ era

NISQ — TODAY

Physical qubits run bare. Errors accumulate with every gate, capping useful circuit depth at a few thousand operations.

Heroic calibration, shallow circuits, statistical error mitigation.

FAULT-TOLERANT — THE TARGET

Logical qubits encoded across many physical qubits. Active error correction detects and repairs faults faster than they accumulate.

Arbitrarily deep circuits — the regime where the known exponential speedups live.

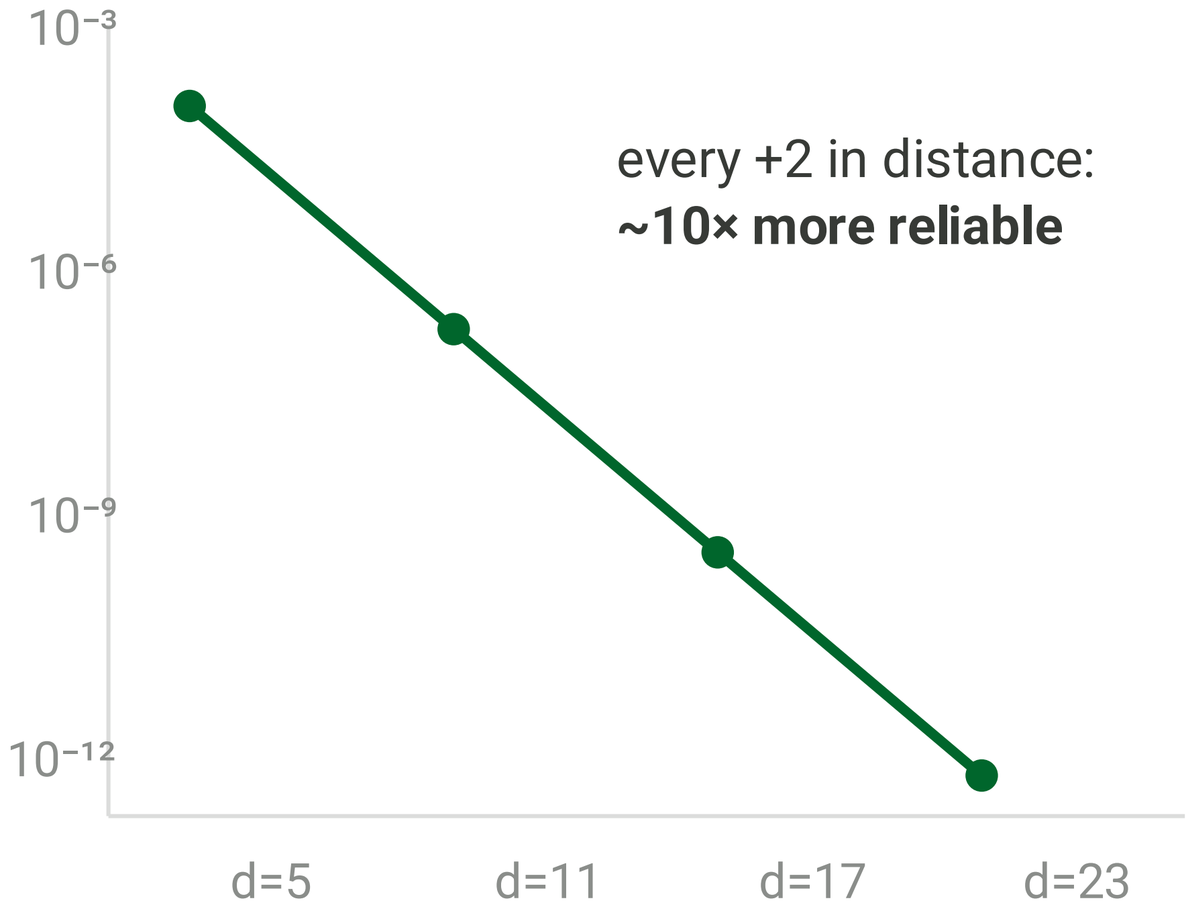
2024–25: below-threshold surface-code operation and logical processing on neutral-atom arrays moved fault tolerance from theory to engineering.

Fault tolerance trades physical qubits for logical reliability

$$P_L = a\,(p / p^*)^{(d+1)/2}$$

Surface code: logical error rate P_L falls exponentially with code distance d — provided the physical error rate p stays below the threshold p* ≈ 1%.

The price: each logical qubit consumes about **2d² physical qubits** , plus magic-state factories for every non-Clifford gate.



The overhead is staggering — and that is the point

~1,000×

physical qubits per logical qubit at useful code distances

20 M

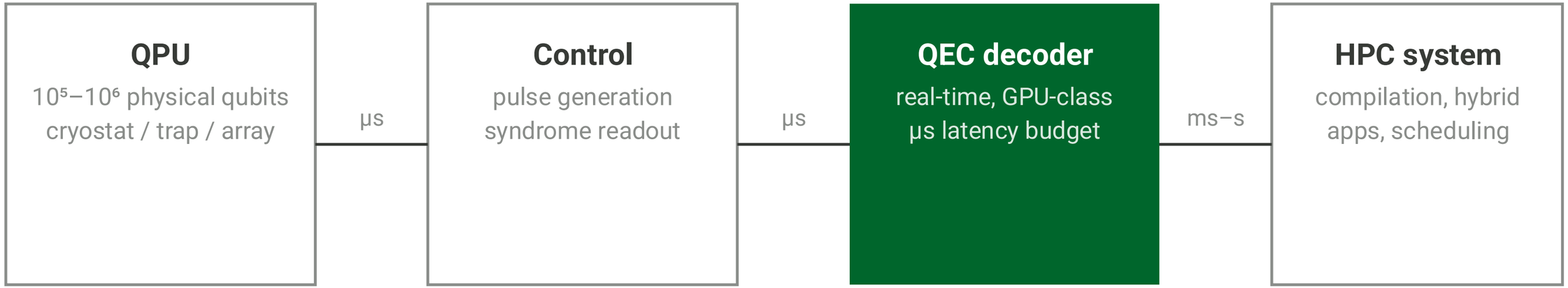
noisy qubits to factor RSA-2048 in 8 hours (Gidney & Ekerå 2021)

122 d

runtime to simulate the FeMo-co catalyst on 1.2×10^8 qubits (Mohseni et al. 2024)

Every design decision — qubit modality, error-correcting code, error budget — moves these numbers by **orders of magnitude**. That sensitivity is a modeling problem.

A fault-tolerant machine is mostly a classical machine



The quantum device is one component. Everything to its right is classical systems engineering — **the territory this community already knows.**

QEC decoding is a real-time HPC workload

Every ~1 μs surface-code cycle emits a syndrome for **every logical qubit** . The decoder must infer and correct errors before the backlog stalls the computation.

At hundreds of logical qubits, that is a continuous stream of graph-matching problems with a **hard microsecond deadline** — a job for GPU-class accelerators sitting next to the cryostat.

Throughput, latency tails, interconnect placement, co-scheduling: the classic vocabulary of this workshop.

684×

decoding-latency reduction from architecture-aware error handling (Kim et al., ASPLOS 2024)

10^{10}×

logical error-rate improvement in the same million-qubit design study

Nobody can yet predict what these systems will cost

Quantum resource estimation (QRE) : given an algorithm, a hardware target, and an error budget — how many physical qubits, how long, at what code distance?

Independent published estimates of the **same computation** differ by an order of magnitude or more, depending on assumptions buried inside each tool.

Hardware roadmaps, algorithm choices, and a 2028 national deadline all hang on these numbers.

FEMO-CO GROUND-STATE ESTIMATES

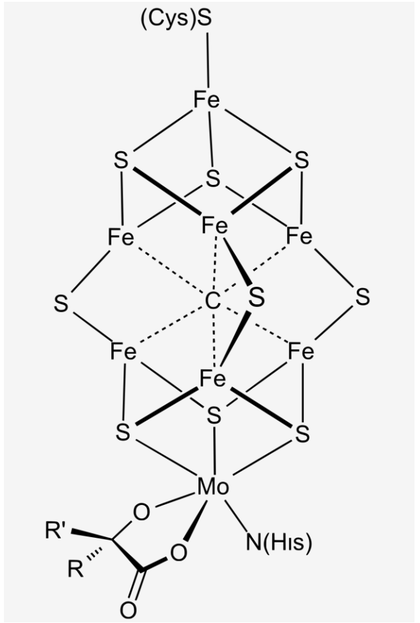
1.2×10^{8} qubits

Mohseni et al. 2024, double factorization

2.1×10^{6} qubits

Kim et al. 2024, distributed architecture, d=31

Same molecule. ~50× apart.



FeMo-co — nitrogenase's FeMo cofactor: the active site where nature fixes N_2 at room temperature. Understanding it could remake fertilizer production (~2% of world energy). Source: Wikipedia

Today's estimation tools don't compose

Capability	Azure QRE	Qualtran	pyLIQTR	MQT
Logical estimation	●	●	●	—
Physical estimation	●	—	—	●
Custom QEC codes / hardware	—	—	—	—
Portable interchange between tools	—	—	—	—

Different gate sets, QEC models, and error-budget defaults behind incompatible APIs. Disagreements can't be diagnosed — and upstream bugs propagate **silently** into published numbers.

This is the co-design problem

Interchange formats — LLVM IR, ONNX, MLIR	→ **A portable logical-resource artifact**
Analytical cost models of architectures	→ **QEC overhead and factory models**
Design-space exploration of HPC systems	→ **Hardware × code × error-budget sweeps**
Multi-objective optimization in co-design	→ **Pareto fronts over qubits and runtime**

And vendors will always keep proprietary tools. **Good interfaces are what let open and vendor stacks coexist** : comparable numbers for procurement, cross-tool debugging, portable application development.

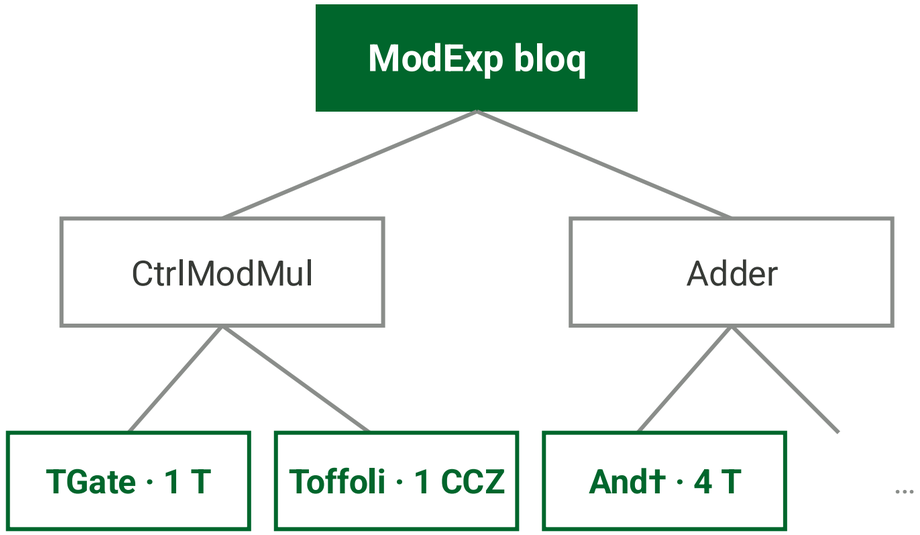
Decades of methodology transfer directly. The rest of this talk: our first pass at applying it.

We built a two-stage estimation pipeline around a frozen interchange format

Any producer pairs with any consumer — **including proprietary vendor tools behind the same interface** . Strict mode refuses silent backend substitution; every reported number names the producer–consumer pair that made it. That's what makes estimates comparable across procurements, disagreements debuggable, and applications portable.

Stage 1 exemplar: Qualtran

Google · logical decomposition



get_cost_value(bloq, QECGatesCost()) walks the graph, memoizing per (bloq, key) — repeated primitives cost one visit.

An algorithm is a **hierarchical graph of "bloqs"** — each declares a register signature and either a leaf cost or a decomposition into callees.

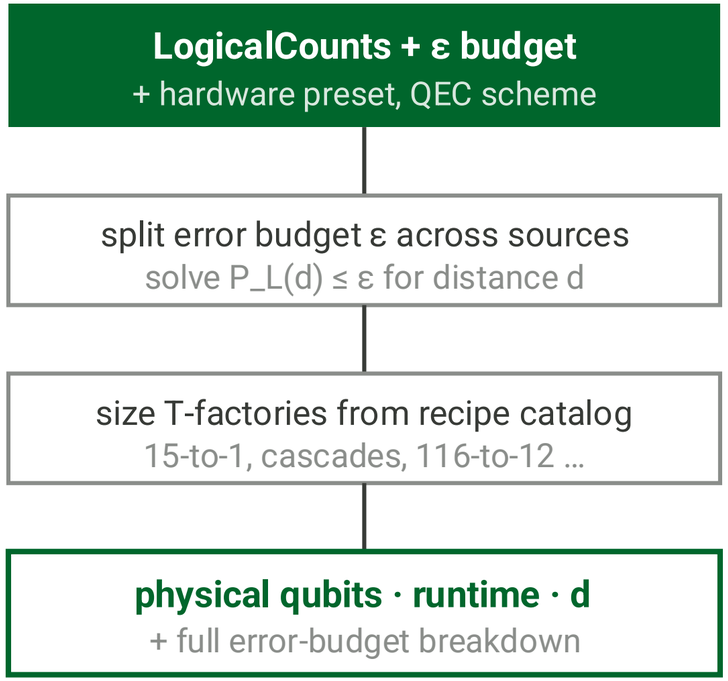
Costs compose upward: leaf contributions weighted by caller multiplicity. A chemistry circuit invoking one primitive a million times pays the recursion **once**.

Logical-only by design — no QEC model, no hardware. Our adapter maps algorithm templates to bloqs and folds the counts into the interchange format.

Harrigan, Babbush et al., *Expressing and Analyzing Quantum Algorithms with Qualtran* , arXiv:2409.04643 · github.com/quantumlib/Qualtran

Stage 2 exemplar: Azure Quantum Resource Estimator

Microsoft · physical estimation



Implements the Beverland et al. model: tiles for logical qubits, O(√n) lattice-surgery routing, factories sized to T-state demand.

The most complete single-tool pipeline: surface and Floquet codes, six hardware presets, built-in **error-budget optimization**.

Our adapter feeds it portable logical counts directly — **bypassing Q# program submission** — so any Stage-1 producer can drive it.

Limits: QEC codes and hardware models outside its presets require patching the tool — the gap FormulaQEC fills.

Beverland, Murali, Troyer, Svore et al., *Assessing requirements to scale to practical quantum advantage* , arXiv:2211.07629 · aka.ms/AQ/RE

Seven numbers are enough to decouple algorithm from hardware

`num_qubits`	`t_count`
`ccz_count`	`rotation_count`
`rotation_depth`	`measurement_count`
`clifford_count`	+ optional circuit_depth

Backend-agnostic by construction: **no gate graph, no source program, no connectivity model, no schedule** . Any producer can emit it; any consumer can accept it.

The omissions are as load-bearing as the inclusions — that is what LLVM IR and ONNX got right.

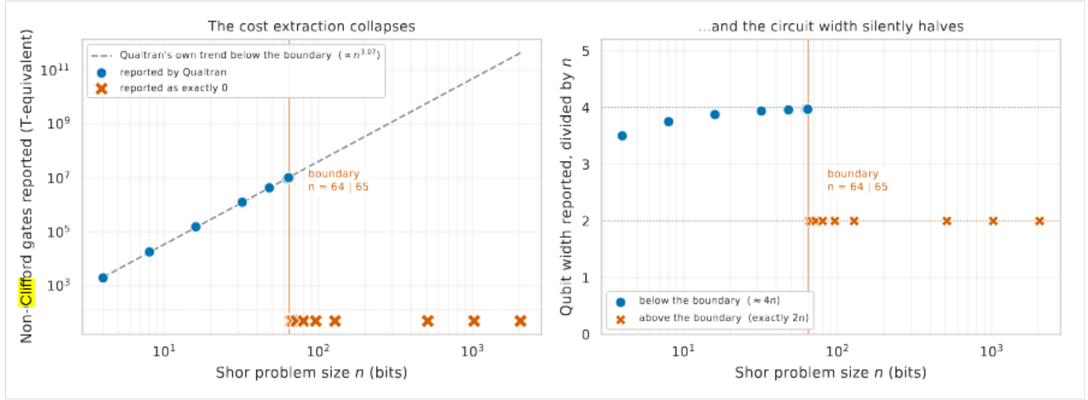
One derived metric normalizes non-Clifford cost: t + 4·ccz + rotations.

Checking invariants at the boundary catches silent tool failures

A production frontend, at large problem sizes, reported **zero non-Clifford gates** for Shor's algorithm — a circuit that plainly requires them. End to end, the result looked like every other estimate.

One format invariant — a non-trivial circuit must have positive T-equivalent cost — **caught it at the handoff** , in code that names no tool.

1	Producer asserts the circuit is non-trivial
2	Substrate checks t + 4·ccz + r > 0 at the stage boundary
3	Violation → diagnostic naming producer and invariant
4	Pipeline refuses to emit a meaningless number



Beyond n = 64, reported non-Clifford cost drops to exactly zero (left) while circuit width silently halves (right) — no exception, no warning.

We reproduce published anchors to within 2.5 percent

Shor-2048 configuration	Published	Ours	Δ
Superconducting, p = 10^{-3}	25.17 M	25.16 M	**0.0%**
Superconducting, p = 10^{-4}	5.83 M	5.98 M	**2.5%**
Majorana, p = 10^{-4}	13.40 M	17.08 M	**27.5%**
Majorana, p = 10^{-6}	4.18 M	5.21 M	**24.6%**

Preset	Modality	Gate time	p_{cliff}
gate_ns_e3	Superconducting	50 ns	10^{-3}
gate_ns_e4	Superconducting	50 ns	10^{-4}
gate_us_e3	Trapped ion	100 μs	10^{-3}
gate_us_e4	Trapped ion	100 μs	10^{-4}
maj_ns_e4	Majorana	100 ns	10^{-4}
maj_ns_e6	Majorana	100 ns	10^{-6}

The six hardware presets behind these anchors.

Superconducting rows establish the pipeline is **correct** ; a cross-producer chemistry check agrees **exactly** on structural metrics, within 5% on physical qubits.

And the Majorana rows? Both sides consumed **identical logical counts** , so the 25–28% gap can only live downstream — it localizes to **Floquet-code parameter assumptions** , not an implementation error.

Holding the logical artifact fixed turns cross-tool spread into attributable modeling differences — the same move performance modeling made when it separated workloads from machine models.

Hypothetical QEC codes are an equation, not a complete rewrite

```
name: hypothetical-qldpc
qubits_formula: "12 * d"
cycle_time_formula: "6 * t_2q * d"
threshold: 0.01
```

Formulas evaluated by AST walking against a whitelist — safe to share, no code to write. QEC research moves faster than tooling; this keeps estimates current.

Swapping $2d^2$ for $12d$ in one line shows an **order-of-magnitude** qubit reduction ripple through a full system estimate.

Code family	Qubits / logical	Character
Surface / rotated surface	$2d^2$	the workhorse: 2D-local, ~1% threshold, best-understood decoders
Color	$\sim 4.5d^2$	transversal Clifford gates — cheaper logic, costlier decoding
Floquet / Hastings–Haah	$\sim 4d^2$	two-qubit measurements only — suits Majorana hardware
qLDPC (bivariate-bicycle, lifted-product)	~12d — near-linear	IBM: 12 logical in 144 physical — needs long-range coupling

The design space is wide open — and mostly unexplored

PRELIMINARY

Full-factorial sweep

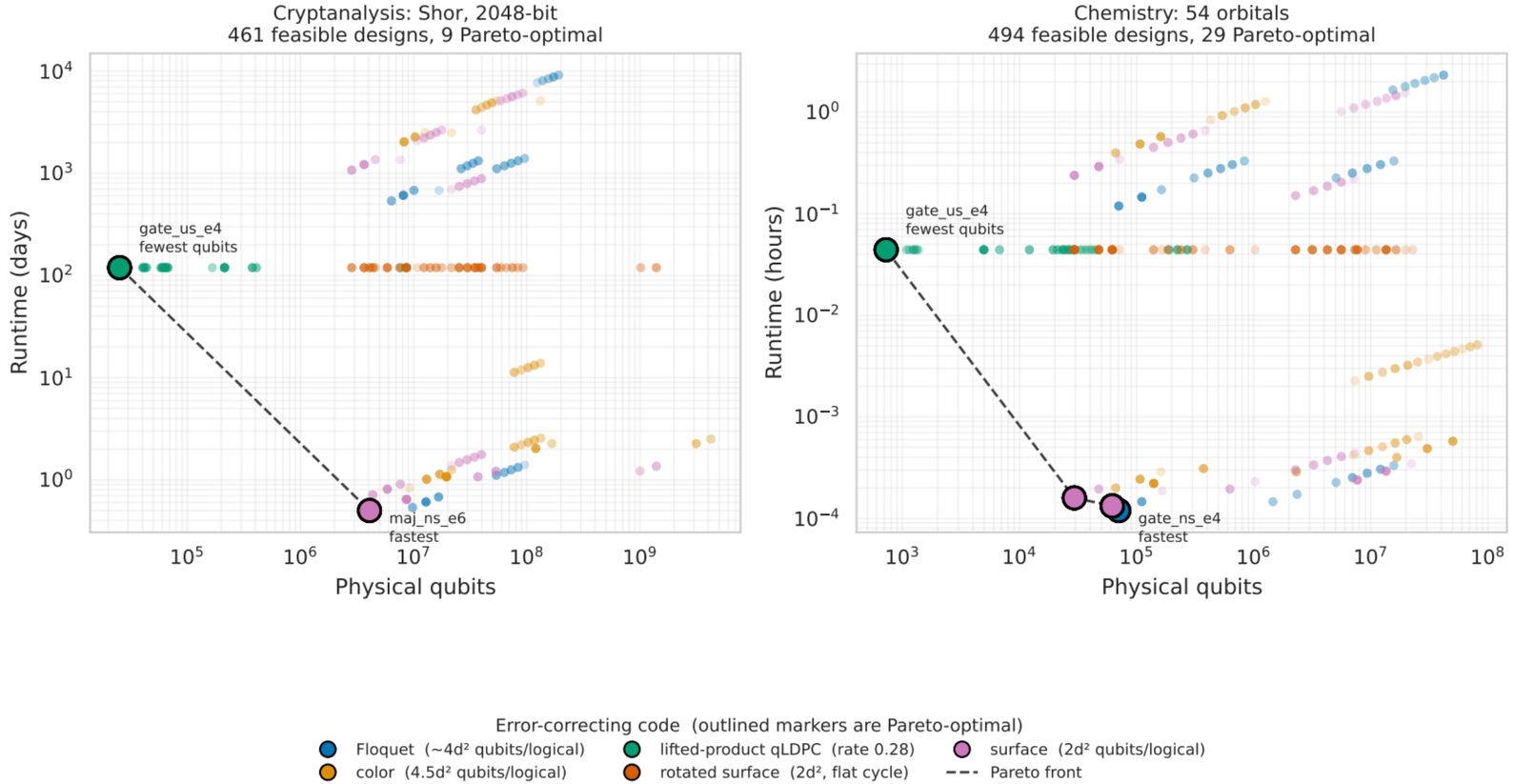
hardware presets × 5 QEC codes × error budgets, per application

~460–490 feasible designs each; only 9–29 Pareto-optimal

qubits–runtime trade-offs span 4+ orders of magnitude

Next: OFAT sensitivity + NSGA-II

asymmetric ε splits a discrete grid would miss



Qubits vs. runtime across five QEC codes; outlined markers are Pareto-optimal. Left: Shor-2048. Right: 54-orbital chemistry.

Call to action for the ModSim community

Resource estimation is just the problem we started with — the rest of the HPC–QC co-design stack is equally open:

01 **Energy models** for FTQC — today there are none, at any fidelity

02 **Validation without hardware** — calibrating models for machines that don't exist yet

03 **Sizing QEC decoder systems** — throughput, latency, placement of real-time decoding on accelerators

04 **Application fitness** — which workloads actually merit FTQC, at what problem sizes

05 **Co-scheduling HPC and quantum** — batch and runtime scheduling across both resource types

06 **Distributed FTQC** — resource models spanning multiple cryogenic modules

07 **Community interchange formats** — standards for logical-resource artifacts, with invariants

08 **Closing the loop** — estimates feeding HPC–QC schedulers and runtimes directly

Time to dig in!

Quantum is the richest open frontier for ModSim methodologies in recent memory

Fault tolerance is now an engineering problem — on a 2028 national deadline.

Its physical costs are acutely design-sensitive, and today's estimates are ad hoc.

Interchange formats, cost models, DSE, and MOO transfer directly.

OAK RIDGE National Laboratory

Jeffrey S. Vetter · vetter@ornl.gov

Advanced Computing Systems Research, Oak Ridge National Laboratory