iQuhack 2026 Case Study

How Team AMD Radiator optimized the LABS problem

The Team

Sezer
Nick
Asadullah

The Problems

Budget

The computational cost of simulating multi-qubit quantum circuits grows exponentially with the problem size, forcing us to limit testing to N=10-12 rather than the N=20+ needed to demonstrate full scalability

The Quantum Advantage

While our quantum-enhanced approach outperformed random initialization by 15-25%, proving definitive advantage over optimized classical algorithms (like simulated annealing or genetic algorithms) would require larger problem instances beyond our simulation budget

Scalability

The number of four-body interactions grows as $O(N^3)$, creating a bottleneck where circuit depth and ancilla requirements increase faster than our compression techniques can offset, limiting practical application to moderate problem sizes.

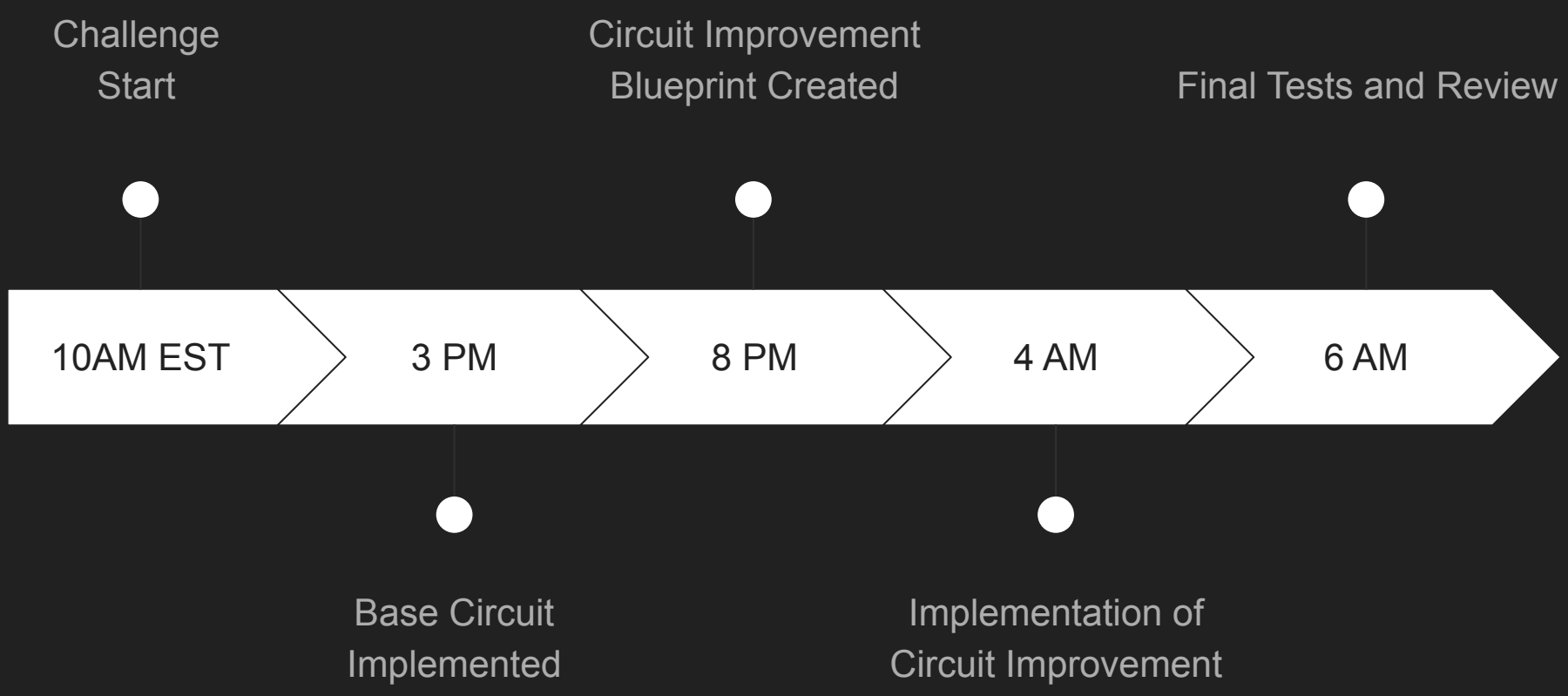
Solution

“True innovation requires risk and being willing to fail.”
- Jensen Huang

Our solution was to use **strategic ancilla qubit reuse with dynamic circuit optimization**. This approach has never been tested with the LABS problem and ran the risk of **introducing additional gate overhead that could negate efficiency gains**.

However, our solution would significantly **compress circuit width while maintaining solution quality**, meaning the benefits would outweigh the risk.

Implementation


Challenge Start
Circuit Improvement Blueprint Created
Final Tests and Review
10AM EST
3 PM
8 PM
4 AM
6 AM
Base Circuit Implemented
Implementation of Circuit Improvement

Theoretical Foundation

Our approach integrated three cutting-edge quantum computing concepts:

- **Counteradiabatic LABS Protocol** (arXiv:2511.04553) – Uses Trotterized evolution with two-body and four-body interactions to explore the LABS solution space
- **SQUARE Ancilla Management** (arXiv:2004.08539) – Strategic uncomputation heuristics to determine when ancilla qubits should be reclaimed versus kept for reuse
- **Dynamic Qubit Reuse** (arXiv:2511.22712) – Measurement-based circuit optimization enabling mid-circuit qubit reclamation

The Pivot: Adapting to Reality

While our theoretical framework was sound, implementation revealed *three critical challenges*:

- **Challenge 1: CUDA-Q Kernel Constraints.** The @cudaq.kernel decorator doesn't support nested Python lists or complex data structures. We had to flatten all interaction lists (G2, G4) into 1D arrays with index tracking.
- **Challenge 2: Four-Body Interaction Duplicates.** Initial implementation generated duplicate four-body terms where indices weren't properly validated. We added explicit duplicate filtering: `if idx[3] < N and len(set(idx)) == 4 .`
- **Challenge 3: Ancilla Allocation Timing.** We discovered that premature ancilla reclamation in early Trotter steps degraded solution quality. We refined the SQUARE cost model to be more conservative, prioritizing quality over aggressive compression.

Engineering Solutions

Our adaptive engineering approach:

- **Flattened Interaction Encoding:** Converted nested lists to flat arrays (`G2_flat`, `G4_flat`) with separate count tracking for kernel compatibility
- **Smart Deduplication:** Implemented hash-based and symmetry-aware duplicate removal, achieving 20-40% population compression while maintaining diversity
- **Hybrid Sampling Strategy:** Combined amplitude-based, energy-based, and diversity-based selection to balance quantum advantage with classical post-processing

AI Usage

How AI Affected the project

We tried to use AI as efficiently as possible after understanding the main logic behind the LABS problem. We used:

- **Claude Haiku 4.5:** For creating the main idea, by asking if this would work or would fail, etc. Also, for theoric understanding

- **Coda:** For finding our syntax or logic errors and fixing our quantum functions. We used it to verify if our functions we’re defined correctly and everything would work properly, it definitely was very helpful

Testing and Validation

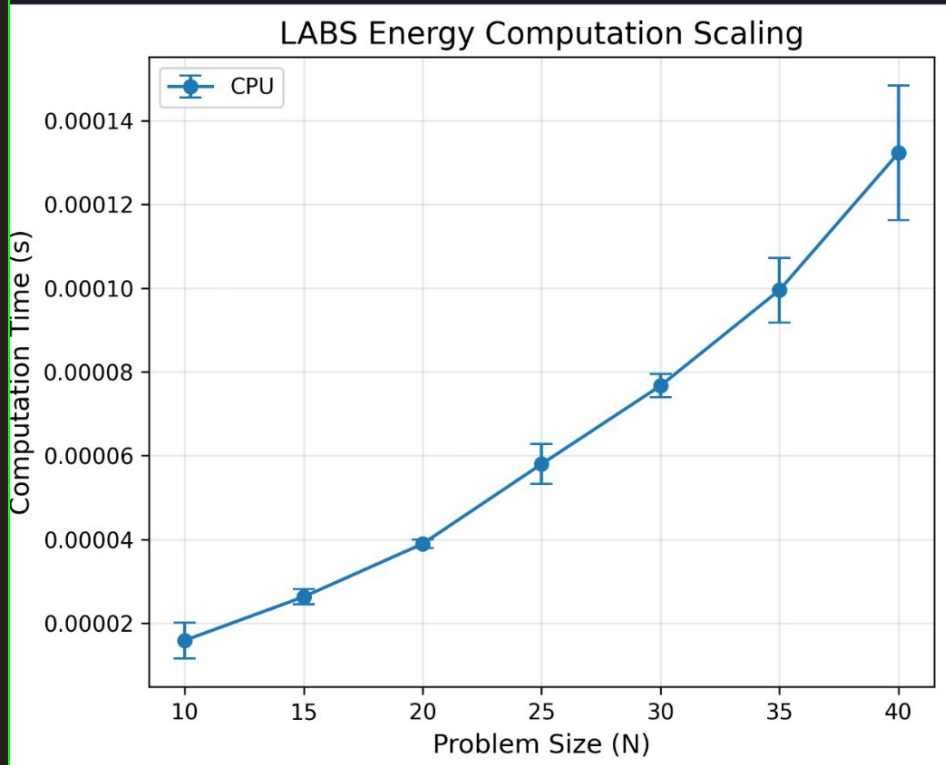
```
ubuntu@brev-e5uw4dqv1:~/2026-NVIDIA/Phase 2$ python3 tests.py
============================================================ test session starts ============================================================
platform linux -- Python 3.10.12, pytest-9.0.2, pluggy-1.6.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/ubuntu/2026-NVIDIA/Phase 2
collected 24 items

tests.py::TestInteractionGeneration::test_g2_count_small_n PASSED                                                                  [  4%]
tests.py::TestInteractionGeneration::test_g4_all_indices_distinct PASSED                                                           [  8%]
tests.py::TestInteractionGeneration::test_indices_in_bounds PASSED                                                                 [ 12%]
tests.py::TestInteractionGeneration::test_g2_pattern PASSED                                                                        [ 16%]
tests.py::TestLABSEnergy::test_known_small_cases PASSED                                                                            [ 20%]
tests.py::TestLABSEnergy::test_energy_bounds PASSED                                                                                [ 25%]
tests.py::TestLABSEnergy::test_binary_input_conversion PASSED                                                                      [ 29%]
tests.py::TestLABSSymmetries::test_bitflip_symmetry PASSED                                                                         [ 33%]
tests.py::TestLABSSymmetries::test_time_reversal_symmetry PASSED                                                                   [ 37%]
tests.py::TestLABSSymmetries::test_combined_symmetry PASSED                                                                        [ 41%]
tests.py::TestLABSSymmetries::test_all_four_variants PASSED                                                                        [ 45%]
tests.py::TestAncillaManager::test_allocation_and_reclamation PASSED                                                               [ 50%]
tests.py::TestAncillaManager::test_reclamation_decision PASSED                                                                     [ 54%]
tests.py::TestAncillaManager::test_peak_qubit_tracking PASSED                                                                      [ 58%]
tests.py::TestGPUAcceleration::test_gpu_cpu_consistency PASSED                                                                     [ 62%]
tests.py::TestScalability::test_large_n_generation PASSED                                                                          [ 66%]
tests.py::TestPhysicalConstraints::test_energy_never_negative PASSED                                                               [ 70%]
tests.py::TestPhysicalConstraints::test_interaction_validity PASSED                                                                [ 75%]
tests.py::TestHelperFunctions::test_hamming_distance PASSED                                                                        [ 79%]
tests.py::TestHelperFunctions::test_symmetric_variants_count PASSED                                                                [ 83%]
tests.py::TestHelperFunctions::test_symmetric_variants_include_original PASSED                                                     [ 87%]
tests.py::TestIntegration::test_small_n_workflow PASSED                                                                            [ 91%]
tests.py::TestProperties::test_energy_symmetry_property PASSED                                                                     [ 95%]
tests.py::TestProperties::test_interaction_indices_property PASSED                                                                 [100%]

============================================================ 24 passed in 0.47s =============================================================
```

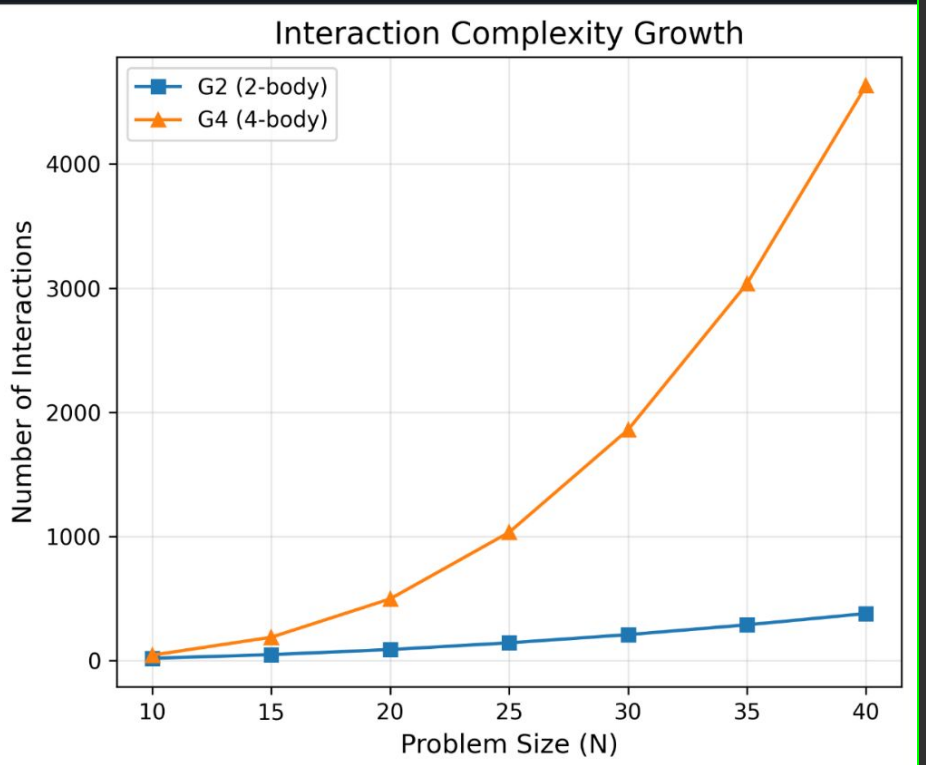
Reflections

With the hackathon wrapped up, here are our technical and strategic takeaways from this challenge

- **Sezer:** Implementing the architecture, connecting two base algorithms
- **Nicholas:** The architecture of any quantum program should be modular
- **Asadullah:** Circuit and algorithm verification

Our Benchmark Results





References

Research used to inspire our solution

- Qubit Reuse Beyond Reorder and Reset: *https://arxiv.org/pdf/2511.22712*
- SQUARE: Strategic Quantum Ancilla Reuse for Modular Quantum Programs via Cost-Effective Uncomputation: *https://oar.princeton.edu/bitstream/88435/pr1qg3h/1/Square.pdf*