BOSS: Blocking Algorithm for Optimizing Shuttling Scheduling in Ion Trap

Xian Wu, Chenghong Zhu, Jingbo Wang, Xin Wang

HKUST (GZ) and BAQIS

April 2, 2025

Introduction: Ion Traps and Shuttling

- Ion traps offer high-fidelity gates, long coherence, and full connectivity.
- Scalability is limited by heating and control complexity with more ions.
- Shuttling ions using microelectrodes allows fixed-laser execution zones.
- Reducing shuttle operations is key to fidelity and runtime.

Background: TILT Architecture

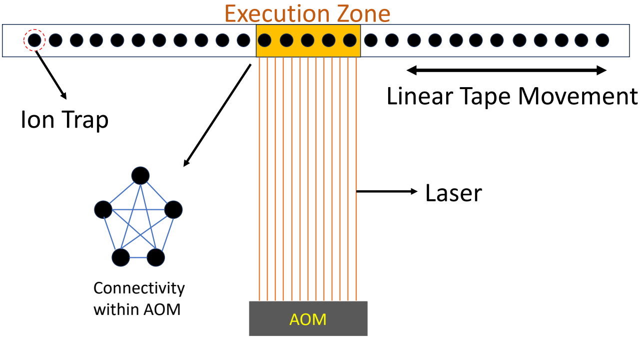

Figure: Linear trapped-ion quantum computer with full connectivity inside the AOM zone. Outside the zone, operations require ion shuttling. In TILT, each shuttle enables arbitrary-distance tape movement.

Background: Another Similar Architecture

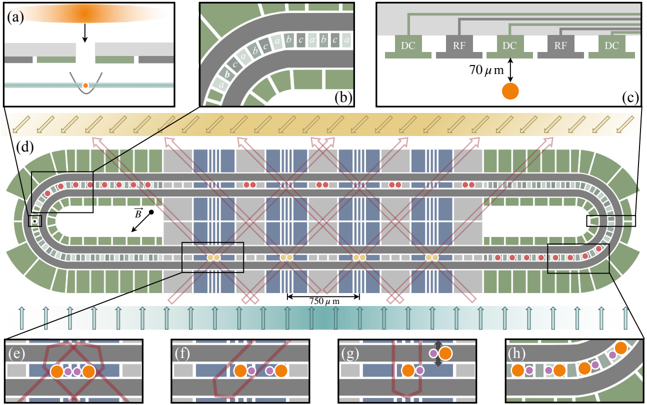

Figure: An overview of the H2 ion trap designed by Quantinuum

Motivation: Optimize Shuttling

- Prior compilers insert SWAPs heuristically without minimizing shuttles.
- Excessive shuttles lead to fidelity loss and longer circuit runtime.
- Opportunity: Use ion trap full connectivity in fixed zones efficiently.
- Need for a better compilation strategy tailored to TILT architecture.

Methods: BOSS Workflow Overview

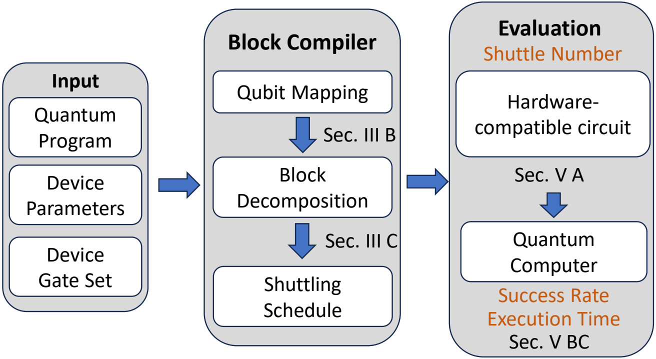

Figure: Inputs: Quantum circuit, hardware specs, native gates. Outputs: Compiled schedule with minimized shuttle operations.

Methods: Circuit Blocking

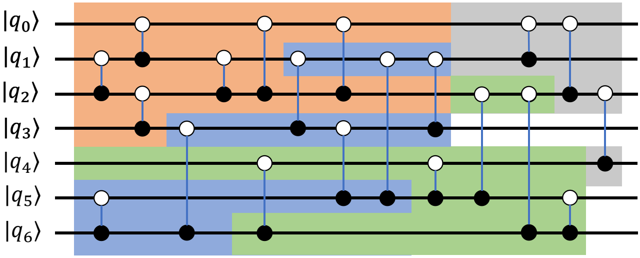

Figure: An example of circuit blocking, the maximal block size is 4. Gates are grouped into blocks respecting the AOM zone size. More gates per block $\Rightarrow$ fewer shuttles.

Methods: Block Scheduling Strategy

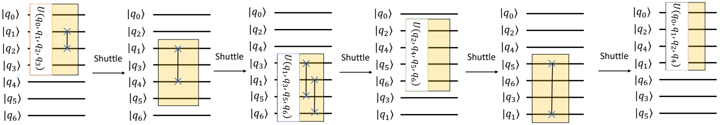

Figure: Example of block scheduling: a block executes when all required qubits are in the execution zone; otherwise, shuttling and SWAP gates are used to gather them.

Overall Results

Application	AOM size	$T_{pre}(s)$	$T_{boss}(s)$	S_{base}	S_{pre}	S_{boss}	ΔS	$\Delta S/S_{pre}$	$t_{pre}(s)$	$t_{boss}(s)$	t_{pre}/t_{boss}
Adder	16	6.216	0.007	35	10	8	2	20%	2.967	0.0378	78.6
BV	16	1.579	0.006	4	4	4	0	0%	0.856	0.0354	24.2
QAOA	16	2.761	0.038	113	18	18	0	0%	1.564	0.0418	37.4
ALT	16	2.880	0.013	91	24	20	4	16.7%	1.311	0.0256	51.2
RCS	16	3.149	0.045	277	65	106	-41	-63.1%	1.704	0.2057	8.3
QFT	16	64.823	0.064	407	162	48	114	70.4%	24.820	0.4405	56.3
SQRT	16	131.245	0.430	816	168	71	97	57.7%	46.554	0.2593	179.6
Adder	32	3.250	0.006	5	5	4	1	20%	3.252	0.0372	87.5
BV	32	0.902	0.005	2	2	2	0	0%	0.987	0.0527	18.7
QAOA	32	1.112	0.029	40	4	4	0	0%	1.357	0.0284	47.7
ALT	32	1.404	0.019	38	8	5	3	37.5%	1.017	0.0178	57.1
RCS	32	0.681	0.017	86	11	21	-10	-90.9%	0.856	0.1307	6.5
QFT	32	37.341	0.051	69	69	8	61	88.4%	33.876	0.3926	86.3
SQRT	32	56.309	0.013	431	76	3	73	96.1%	40.817	0.2070	107.1

Table: Comparison with prior work on compilation time, number of shuttle operations, and total execution time (lower is better). The method achieves up to a 96.1% reduction in shuttling (SQRT benchmark) and an average reduction of 16.6%. T: compilation time; S: shuttle number; t: execution time(Trout model).

Results: Shuttle Number

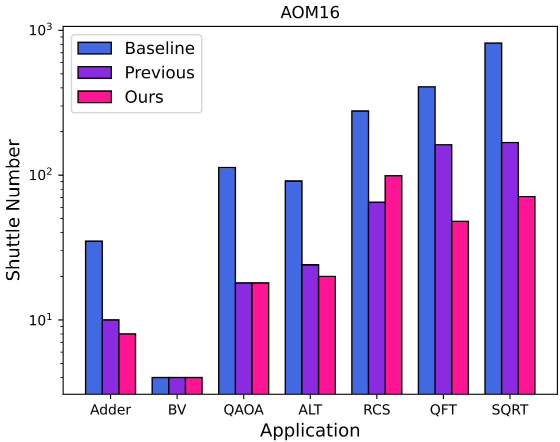

Figure: Number of shuttling operations.

Results: Fidelity

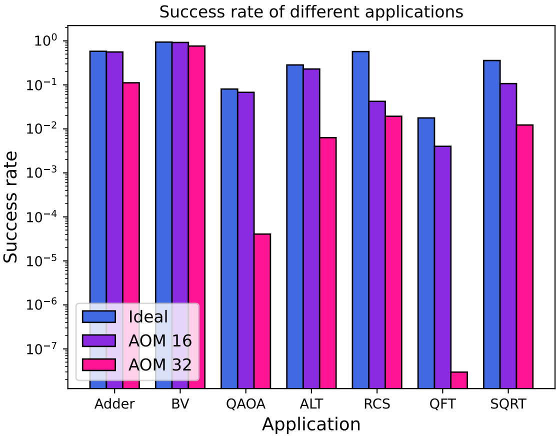

Figure: Orders-of-magnitude improvement in success rate due to fewer shuttles.

Results: Execution Time

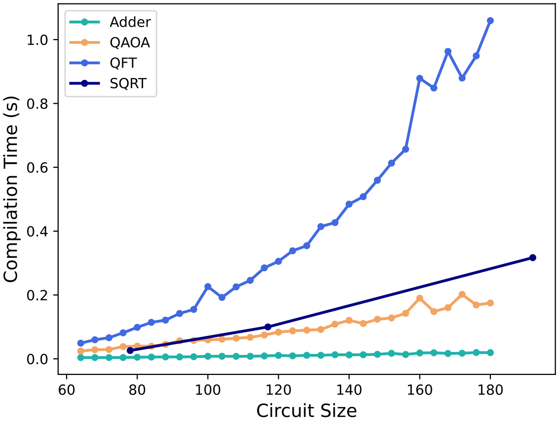

Figure: The compilation time varies with the size of the circuit. The circuit size refers to the number of qubits in the circuit, and the AOM size is 16.

Conclusion

- BOSS reduces shuttling and improves fidelity and runtime.
- Efficient: Compiles large circuits (> 100 qubits) in under 2s.
- Scalable and adaptable to future ion trap designs.