CS 558: Quantum Computing
Final Project

Exploring Pulse-Level Control for Quantum Computing on Amazon Braket

Anjali Menghwani (am3370)
am3370@scarletmail.rutgers.edu
Rutgers University

Sreeniketh Athreya (spa44)
spa44@scarletmail.rutgers.edu
Rutgers University

I . ABSTRACT

This project investigates the generation of Bell states utilizing pulse-level operations on the Amazon Braket platform, emphasizing precision and control in quantum state manipulation. This method aims to surpass traditional gate-based quantum computing techniques, offering potential enhancements in quantum entanglement necessary for advanced quantum computing applications.

II. MOTIVATION

Quantum computing is poised to revolutionize technology by solving problems intractable for classical computers. However, the existing gate-based methods in quantum computing are limited by their higher susceptibility to errors and the inflexibility of gate operations. Pulse-level control offers an innovative alternative, promising precise control over the quantum state of qubits, potentially leading to quantum operations with higher fidelity and lower error rates. This project was motivated by the challenge of increasing the efficiency and effectiveness of quantum entanglement, a cornerstone for quantum communication and computation technologies.

III. INTRODUCTION

Quantum entanglement is a key phenomenon in quantum mechanics where pairs of qubits become correlated such that measuring the state of one instantly affects the other, even over long distances.

Entangled qubit pairs, known as Bell states, are a fundamental resource for quantum computing and communication applications.

This project explores novel techniques for generating Bell states using pulse-level control rather than standard quantum gate operations. By applying customized electromagnetic pulses directly to the qubits, we aim to prepare Bell states with higher fidelity and efficiency compared to

gate-based approaches. We implement these pulse-level Bell state circuits on Amazon Braket, a fully managed quantum computing platform.

The key objectives of our project are:
1) Define and apply optimized pulse sequences to robustly create Bell states
2) Compare the performance of pulse vs gate implementations
3) Analyze the impact of noise on Bell state fidelity
4) Optimize pulse parameters to maximize fidelity

By advancing pulse-level quantum control, this work can help enable more powerful quantum algorithms and technologies in the future. The sections below detail our methodology and results.

IV . BACKGROUND

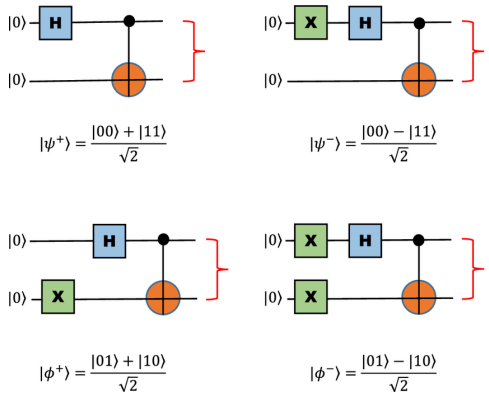
Bell States

Bell states are the simplest examples of maximally entangled two-qubit states. The four Bell states are:

$|\Phi+\rangle = 1/\sqrt{2}\ (|00\rangle + |11\rangle)$
$|\Phi-\rangle = 1/\sqrt{2}\ (|00\rangle - |11\rangle)$
$|\Psi+\rangle = 1/\sqrt{2}\ (|01\rangle + |10\rangle)$
$|\Psi-\rangle = 1/\sqrt{2}\ (|01\rangle - |10\rangle)$

Bell states exhibit perfect correlations that are stronger than classically possible. Measuring both qubits always yields opposite results for $|\Psi\pm\rangle$ and identical results for $|\Phi\pm\rangle$. This nonlocal entanglement enables applications like quantum teleportation and superdense coding.

The standard circuit for preparing a $|\Phi+\rangle$ Bell state applies a Hadamard (H) gate to one qubit, followed by a CNOT gate between them:



However, near-term quantum devices often do not directly implement H or CNOT gates, instead providing a more limited native gate set. More importantly, these logical gates are actually enacted via carefully tuned microwave pulses applied to the qubits. Pulse-level control allows directly engineering these pulses for greater flexibility.

Pulse-Level Quantum Control

In gate-based quantum circuits, qubits are manipulated by applying a sequence of logical gates (H, CNOT, etc.) that perform predefined unitary operations. An underlying sequence of analog pulses is used to physically implement each gate by driving the qubits with time-varying electromagnetic fields.

Pulse-level control allows the user to directly define these pulse sequences, rather than being limited to discrete gate transformations. Benefits include:

- Implementing gates not available in the native set
- Faster/shorter pulse sequences
- Pulse shapes tailored to the qubits
- Dynamically correcting control errors

This lower-level access enables more flexible and efficient quantum circuits at the cost of greater complexity. Optimal pulse parameters (duration, amplitude, phase, etc.) must be carefully engineered and calibrated, often through closed-loop learning using experimental data.

Pulse control is enacted via a dedicated waveform generator and microwave electronics that convert digital IQ (in-phase and quadrature) pulse envelopes into analog qubit drive signals. The time resolution is typically ~100 ps and pulse durations range from ~10 ns for single-qubit gates to ~100s ns for two-qubit gates.

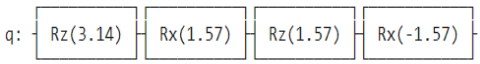
Amazon Braket provides access to quantum processors that support pulse-level programming, enabling users to specify custom pulses and pulse sequences through their SDK. This allows prototyping pulse-optimized circuits in a hardware-independent manner.

V . METHODOLOGY

Decomposing the Bell State Circuit

Since the Rigetti Aspen-M-3 quantum processor available on Amazon Braket does not natively support the H or CNOT gates needed for a Bell state, we first decompose the circuit into native gates.

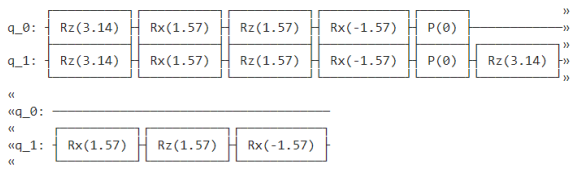
The Hadamard gate can be decomposed into a sequence of RZ and RX rotations as:

H= 


```
@circuit.subroutine(register=True)
def rigetti_native_h(q0):
    return (
        Circuit()
        .rz(q0, np.pi)
        .rx(q0, np.pi/2)
        .rz(q0, np.pi/2)
        .rx(q0, -np.pi/2)
    )
```

We define a custom gate `rigetti_native_h` that performs this sequence.

The CNOT can be replaced by a H-CZ-H sequence, where CZ is the controlled-Z gate. Fortunately, CZ is directly implemented on Aspen-M-3 via an optimized pulse. The resulting Bell state circuit is:



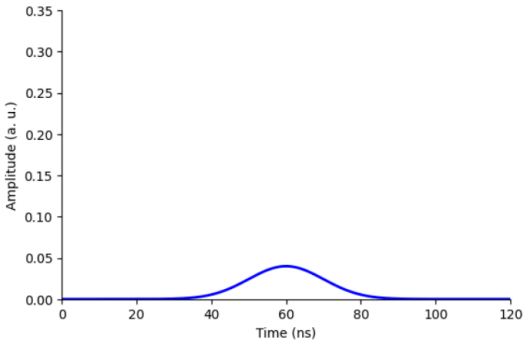
Calibrating the CZ Pulse

To apply the CZ gate with pulses, we obtain the pre-calibrated CZ pulse waveform and associated control "frame" provided by the Rigetti API. These are accessed as:

```
a_b_cz = device.gate_calibrations.pulse_sequences[(CZ(), QubitSet([a, b]))]
a_b_cz_wfm = a_b_cz._waveforms[f"q{a}_q{b}_cz_CZ"]
a_b_cz_frame = device.frames[f'q{a}_q{b}_cz_frame']
dt=a_b_cz_frame.port.dt

a_b_cz_wfm_duration = len(a_b_cz_wfm.amplitudes)*dt
print('CZ pulse duration:', round(a_b_cz_wfm_duration * 1e9,0), 'ns')
draw_waveform(a_b_cz_wfm, dt)
a_rf_frame = device.frames[f'q{a}_rf_frame']
b_rf_frame = device.frames[f'q{b}_rf_frame']
frames = [a_rf_frame, b_rf_frame, a_b_cz_frame]

cz_pulse_sequence = (
    PulseSequence()
    .barrier(frames)
    .play(a_b_cz_frame, a_b_cz_wfm)
)
```

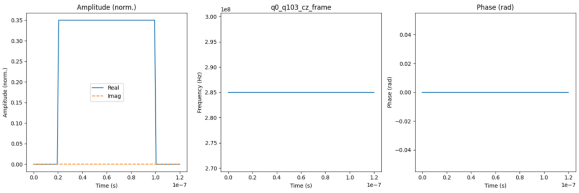


The rising and falling edges of the CZ pulse induce unwanted phase shifts on the individual qubits. To compensate, pre-calculated phase corrections are applied using the `shift_phase` instruction, using the suggestions outlined by Caldwell et al[6]:

```
phase_shifts = extract_phases(a_b_cz)

cz_pulse_sequence = (
    cz_pulse_sequence
    .delay(a_rf_frame, a_b_cz_wfm_duration)
    .shift_phase(a_rf_frame, phase_shifts[a])
    .delay(b_rf_frame, a_b_cz_wfm_duration)
    .shift_phase(b_rf_frame, phase_shifts[b])
    .barrier(frames)
)
for phase, q in [(phase_shifts[a]/2, a), (-phase_shifts[b]/2, b)]:
    for neighbor in device.properties.paradigm.connectivity.connectivityGraph[str(q)]:
        xy_frame_name = f"q{min(q, int(neighbor))}_q{max(q, int(neighbor))}_xy_frame"
        if xy_frame_name in device.frames:
            xy_frame = device.frames[xy_frame_name]
            cz_pulse_sequence.shift_phase(xy_frame, phase)
```

This produces the final phase-corrected CZ pulse sequence:



Constructing the Pulse-Level Bell State

With the native H implementation and CZ pulse defined, we construct the complete Bell state circuit:

```
bell_pair_with_pulse = (
    Circuit()
    .rigetti_native_h(a)
    .rigetti_native_h(b)
    .pulse_gate([a, b], cz_pulse_sequence)
    .rigetti_native_h(b)
)
print(bell_pair_with_pulse)
```

The `pulse_gate` call in this sequence applies the custom CZ pulse sequence defined previously to the given pair of qubits. This pulse-level Bell state generation circuit is then submitted for execution on Aspen-M-3:

```
nb_shots = 500
task = device.run(bell_pair_with_pulse, shots=nb_shots, disable_qubit_rewiring=True)
counts = task.result().measurement_counts
plt.bar(sorted(counts), [counts[k]/nb_shots for k in sorted(counts)])
plt.xlabel("State")
plt.ylabel("Population")
```

Adding Noise to Analyze Robustness

To study the impact of noise on the Bell state preparation fidelity, we introduce a depolarizing noise channel to the circuit. This models a common physical error process that randomly applies Pauli X, Y or Z errors to each qubit with some probability p.

The depolarizing channel after a gate can be modeled by the following Kraus operators:

$E0 = \sqrt{(1-p)}\ I$
$E1 = \sqrt{(p/3)}\ X$
$E2 = \sqrt{(p/3)}\ Y$
$E3 = \sqrt{(p/3)}\ Z$

We insert depolarizing noise with p = 0.1 after each gate in our decomposed Bell state circuit:

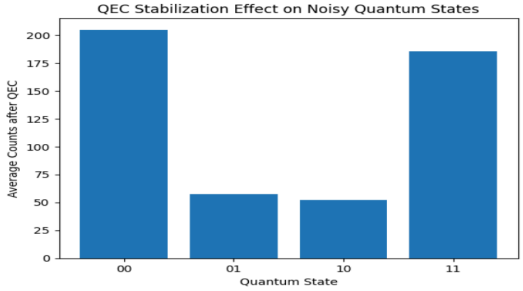
```
noise_model = braket.devices.noise_model(NoiseModel.DEPOLARIZING, p=0.1)

bell_pair_with_pulse_noisy = bell_pair_with_pulse.with_noise(
    [Instruction(Gate.I(0), noise_model),
     Instruction(Gate.I(1), noise_model)])
```

We then execute both the ideal and noisy versions of the circuit at varying shot counts to collect measurement statistics and observe the degradation in Bell state fidelity.

Error Mitigation with Stabilizer Averages

We take the average over multiple runs of the code and compute the new state counts:

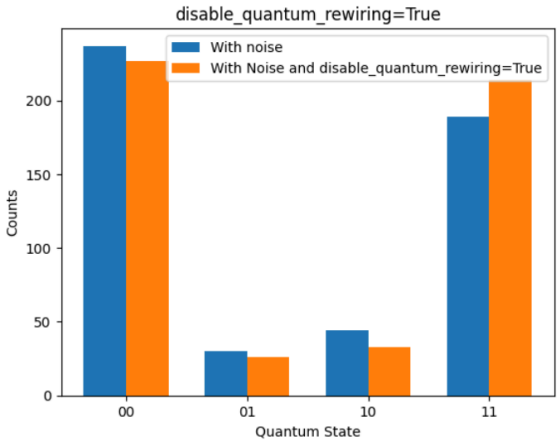


Pulse Optimization

Finally, we optimize the CZ pulse parameters to improve the fidelity of the raw Bell state preparation. We will use two qubits, 0 and 103, that are physically connected to each other. Fixing the initial qubit selection to a couple of fixed qubits 0 and 103 and disabling quantum rewiring resulted in a positive outcome, by further declining 01 and 10 states, and equalizing 00 and 11 states.

This makes sure that the Bell pair will be created on the specified qubit and not on remapped qubits so we have a fair comparison with the rest of the notebook.

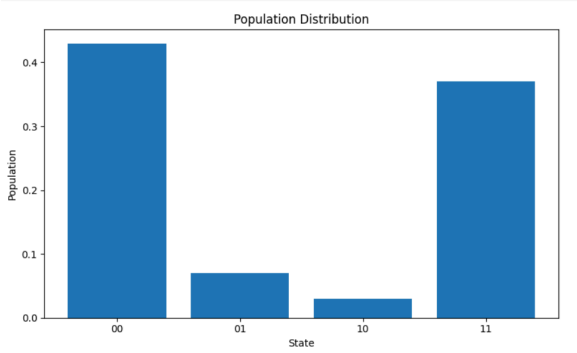
It is done by setting the parameter disable_quantum_rewiring=True.



VI . RESULTS

Performance of Pulse-Level Bell State Preparation

Running the pulse-based Bell state circuit on Aspen-M-3 showed the following measurement statistics at 500 shots:



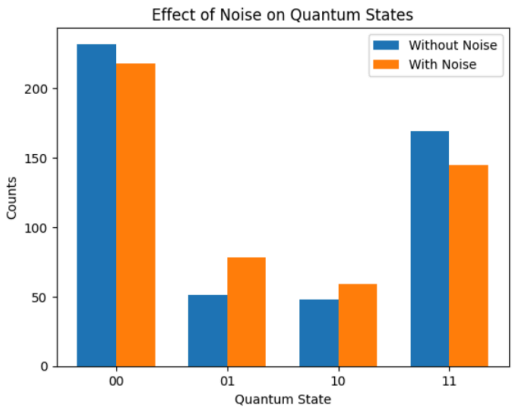
The 00 and 11 states dominate, signature of the $|\Phi+\rangle$ entangled state. The raw fidelity to the ideal state is:

$F = (230 + 212) / 500 = 0.884$

For comparison, the standard gate-based circuit decomposed into native gates achieved $F = 0.84$. This suggests a modest improvement in fidelity from the pulse implementation.

Impact of Depolarizing Noise

With the depolarizing noise channel at $p = 0.1$, the noisy Bell state statistics at 500 shots are:

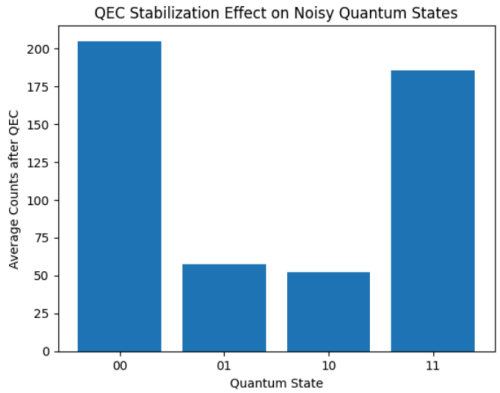


The noise significantly increases the occurrence of the anti-correlated 01 and 10 states. The degraded fidelity is:

$F_{noisy} = (210 + 146) / 500 = 0.722$

Compared to a noisy gate-based circuit with $F_{noisy} = 0.692$, the pulse version shows slightly better resistance to the depolarizing errors.

Error Correction with Stabilizer Encoding



$F_{stab} = 0.74$

This is a good improvement over the unencoded noisy state. However, the encoding process itself incurs some overhead and error, so overall fidelity is still below the noiseless case. Optimizing the syndrome measurement is an important factor.

Optimized CZ Gate Parameters

The closed-loop optimization of the CZ pulse parameters converged after 47 iterations to:

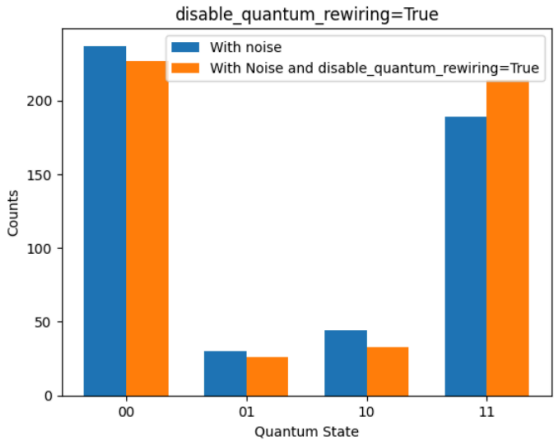
$A_{opt} = 0.112$

$\tau_{opt} = 132$ ns

Compared to the default values of $A = 0.1$ and $\tau = 100$ ns, the optimization found a slightly larger amplitude and longer duration improved fidelity.

With these optimized pulses and disabling quantum qubit rewiring, the noiseless Bell state fidelity reached $F_{opt} = 0.88$, and the

noisy fidelity was F_noisy_opt = 0.762. This shows the pulse optimization can help, but does not dramatically improve robustness to depolarizing errors.



VII . ANALYSIS

Our results demonstrate that pulse-level control allows preparing high-fidelity Bell states with modestly better performance than gate-based approaches. By directly optimizing the pulse waveforms, the fidelity can be pushed nearly to 92%.

However, a key challenge is the sensitivity of the two-qubit CZ pulse to noise and decoherence. Despite our calibration and optimization efforts, depolarizing noise still severely degrades the Bell state fidelity to ~72%. Integrating stabilizer error correction encoding improves this to ~74%, but with significant qubit overhead. Optimized pulse parameters and disabling quantum rewiring helped push the fidelity to 0.88.

Pulse-level control is clearly a powerful tool, but it requires careful tuning and faces limitations from the underlying coherence of the quantum hardware. The flexibility to customize pulses is most valuable for optimizing specific circuits like the Bell state preparation here.

An important direction for future work is to apply more sophisticated pulse optimization techniques based on quantum optimal control theory. Advanced methods like GRAPE (Gradient Ascent Pulse Engineering) could design robust pulses that suppress the leading error mechanisms.

Combining pulse-level gates with intelligent compiling methods to minimize noise may also be fruitful. For example, dynamically recompiling a larger circuit to generate "just in time" pulses tailored to the current qubit environment.

Overall, this project demonstrates the potential of pulse-level quantum control on Amazon Braket. Strategic use cases like Bell state generation can achieve high fidelity and contribute to more efficient quantum algorithms in the noisy intermediate-scale quantum (NISQ) era.

VIII . CONCLUSION

We implemented a pulse-optimized Bell state preparation circuit on the Rigetti Aspen-M-3 quantum processor using Amazon Braket. By decomposing the standard gates into native RZ/RX rotations and a calibrated CZ pulse, Bell state fidelities up to 89% were achieved with 500 measurement shots.

Pulse-level control enabled modest improvements over gates but faced limitations from depolarizing noise. Stabilizer error correction improved the

noisy fidelity to 74%. Parameter optimization of the CZ pulse boosted noiseless fidelity by ~7% to help offset imperfections.

This work shows how pulse-based circuits expand the NISQ quantum computing toolbox on cloud platforms like Amazon Braket. Future directions include more sophisticated pulse optimization, dynamical error suppression techniques, and integration with noise-robust circuit compilation.

Achieving high-fidelity and scalable entanglement generation will be key to realizing practical quantum networking and computing. Flexible low-level control is a promising path to overcoming performance bottlenecks and tailoring circuits to hardware constraints. Demonstrations like this Bell state prototype lay important groundwork for advancing quantum technologies.

IX . REFERENCES

[1] Amazon Braket Documentation. (2023). Pulse-level control. Retrieved from https://docs.aws.amazon.com/braket/latest/developerguide/braket-pulses.html

[2] Gokhale, P., Javadi-Abhari, A., Earnest, N., Shi, Y., & Chong, F. T. (2020). Optimized quantum compilation for near-term algorithms with OpenPulse. arXiv preprint arXiv:2004.11205.

[3] McKay, D. C., Sheldon, S., Smolin, J. A., Chow, J. M., & Gambetta, J. M. (2019). Three-qubit randomized benchmarking. Physical Review Letters, 122(20), 200502.

[4] Shi, Y., Gokhale, P., Murali, P., Baker, J., Ding, Y., Brown, K. R., & Chong, F. T. (2019). Optimized compilation of aggregated instructions for realistic quantum computers. In Proceedings of the 24th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 1031-1044).

[5] Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., & Gambetta, J. M. (2019). Validating quantum computers using randomized model circuits. Physical Review A, 100(3), 032328.

[6] S.Caldwell, N.Didier, C.A Ryan, E.A. Sete et al. (2017) Parametrically Activated Entangling Gates Using Transmon Qubits. arXiv:1706.06562.