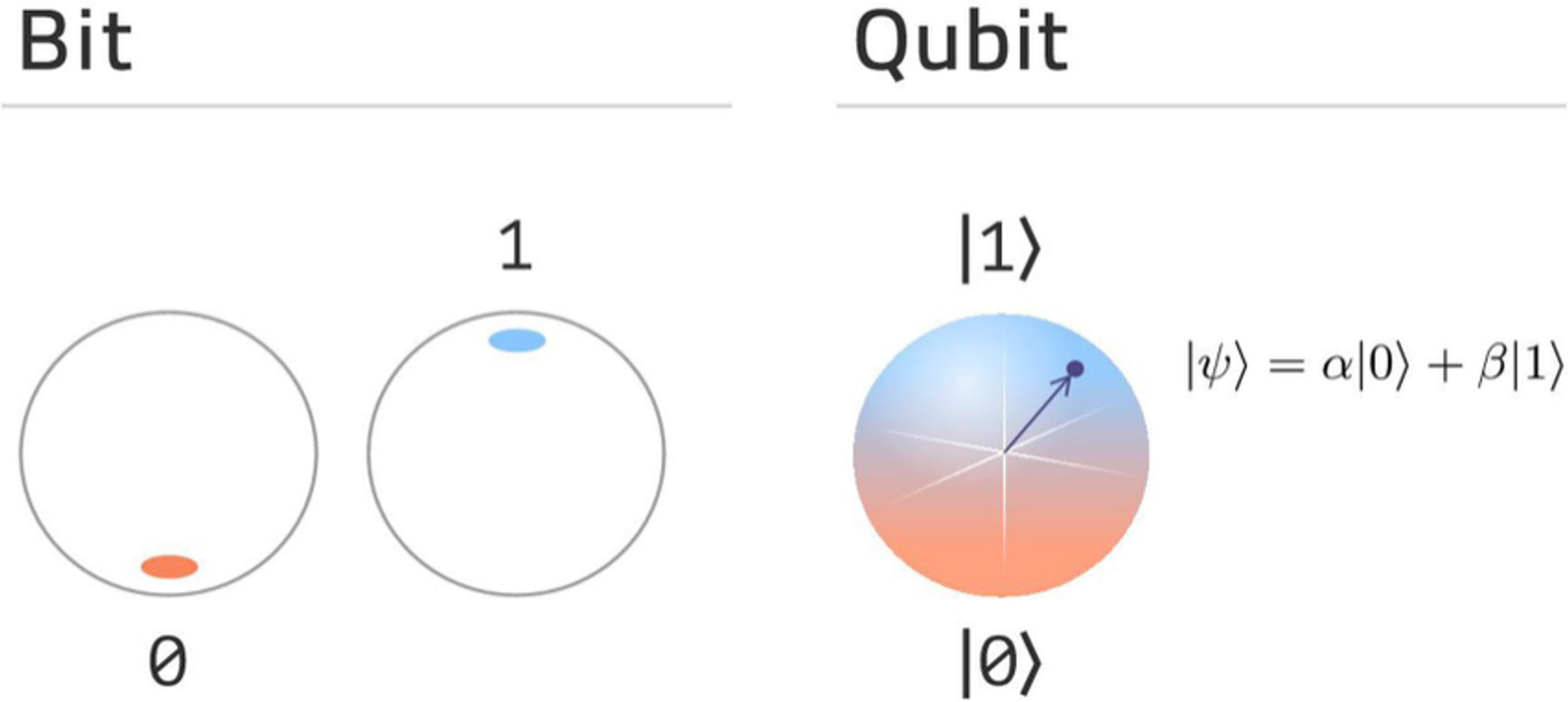
# QIP with NMR: Demonstrating Quantum Advantage

"Willers" Muye yang (Presenter)

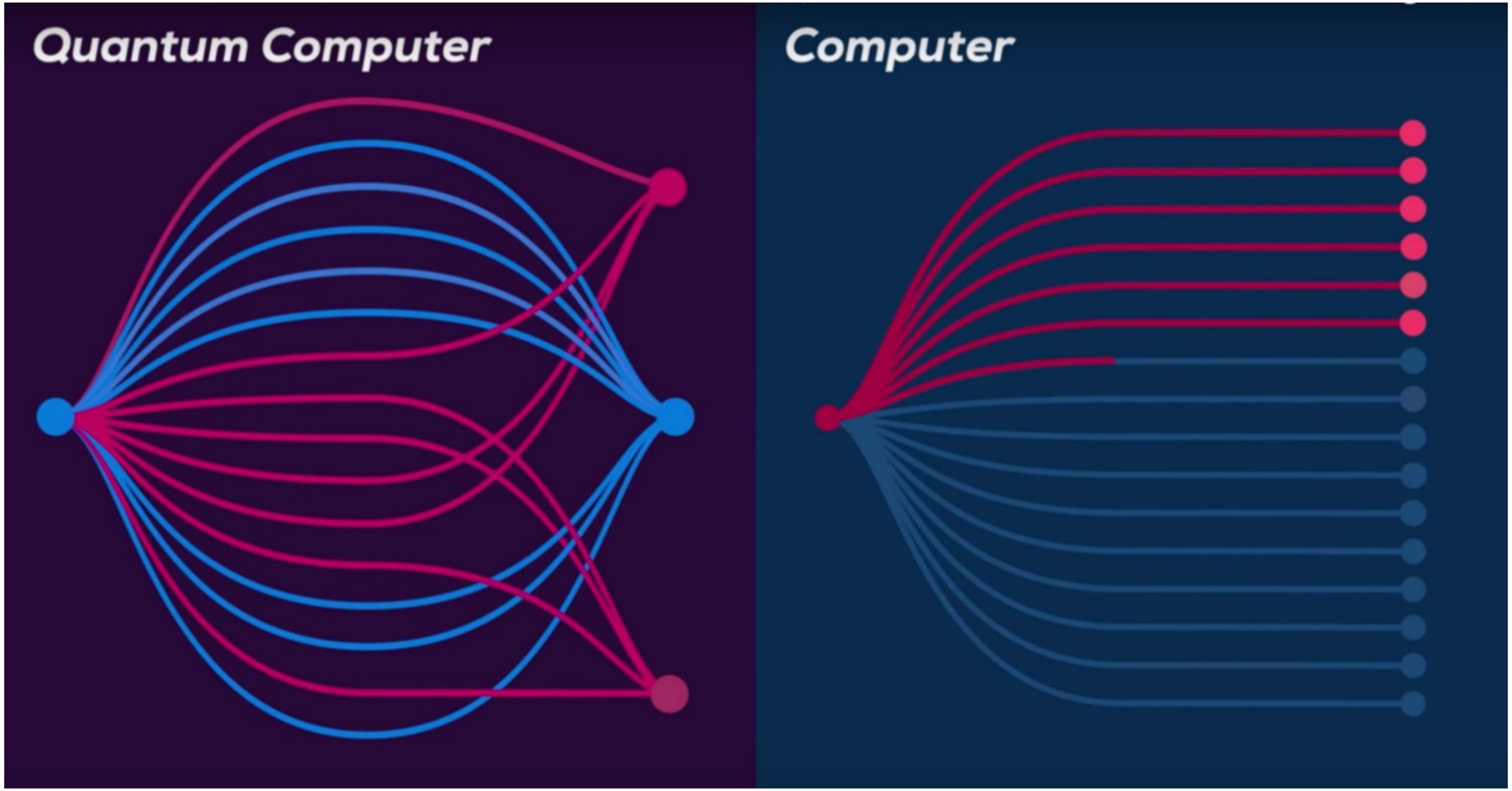
Xiaoyang Zhuang (Lab Partner)

# Beating the Classical Computer

# Quantum Information as Qubits

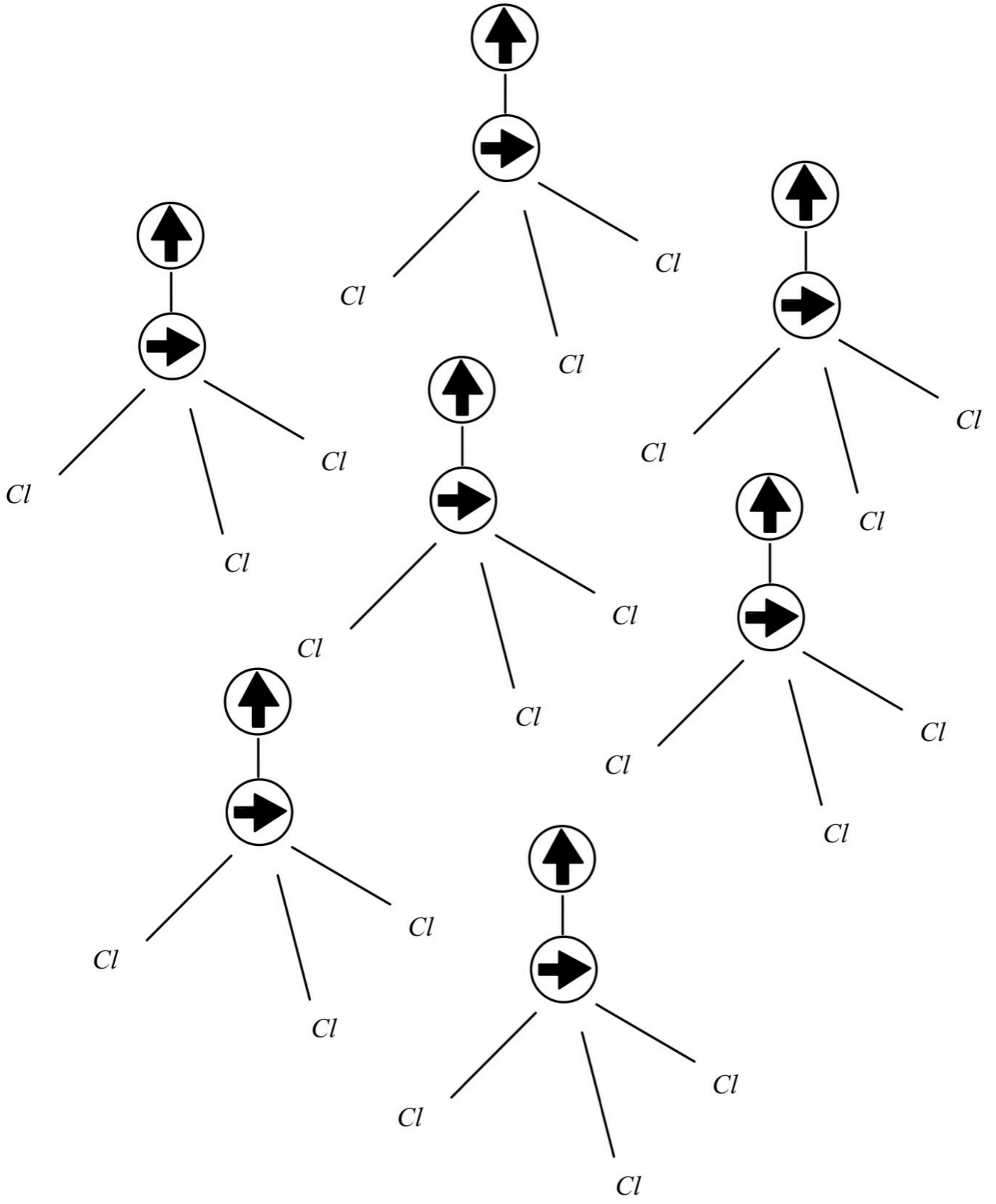


# Quantum Parallelism



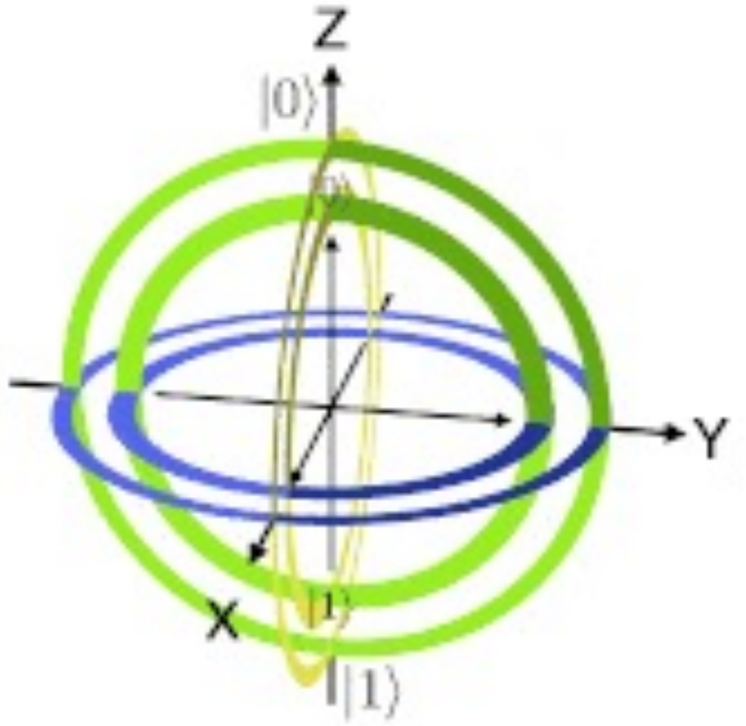
# Realizing Qubits as an Ensemble of Spins

- Our physical qubits are implemented with the magnetic spin of two particles, the Hydrogen nucleus and the Carbon nucleus of $CHCl_3$ denoted
$$|H\rangle \otimes |C\rangle$$
- On the right, the state can be written as
$$|0\rangle \otimes |+\rangle = \frac{|00\rangle + |01\rangle}{\sqrt{2}}$$
- Considers an ensemble of states

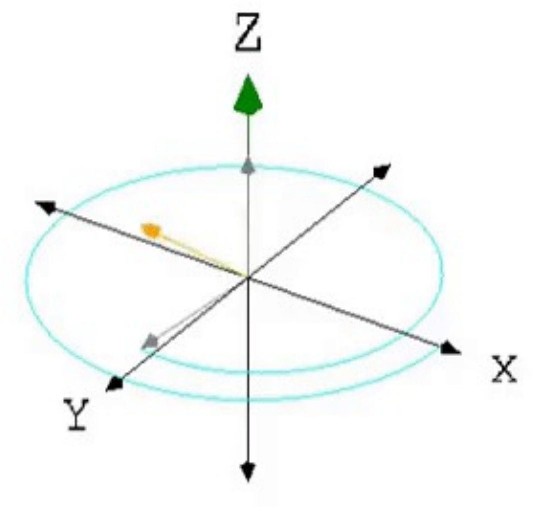


# Realizing Gates as Pulses and Free Evolutions

- Gates, or manipulation of these spin states, are realized via RF pulses.
- Single Rotations:
  - $R_x\left(\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$
  - $R_y\left(\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$
- Free-Evolution:
  - $\tau\left(\frac{1}{2J}\right) = e^{\frac{i\pi}{4}} diag([-i, 1,1, -i])$

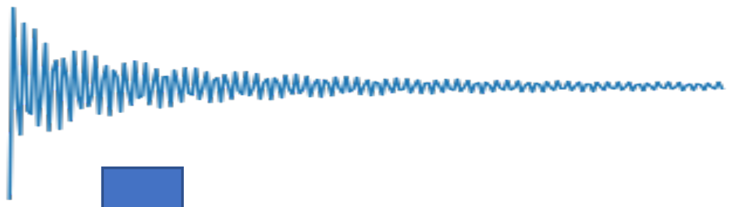


# Realizing Measurements as Spectra of FID

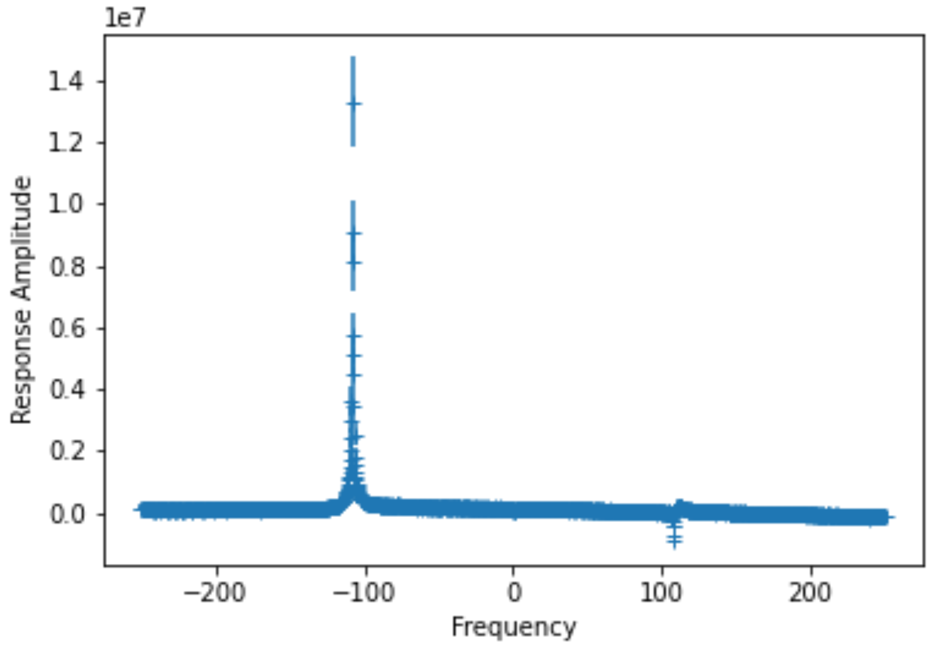



Tyler Moore, 2011

Read out FID for Pure State $|00\rangle$



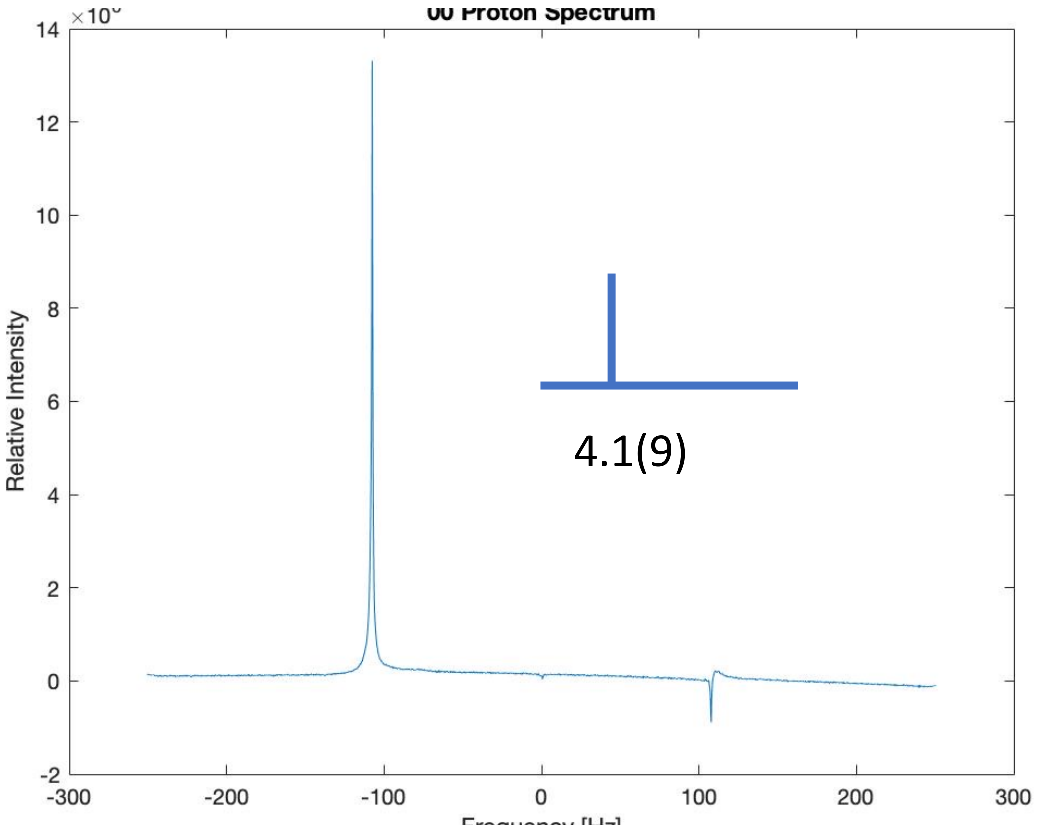
Fourier Transform



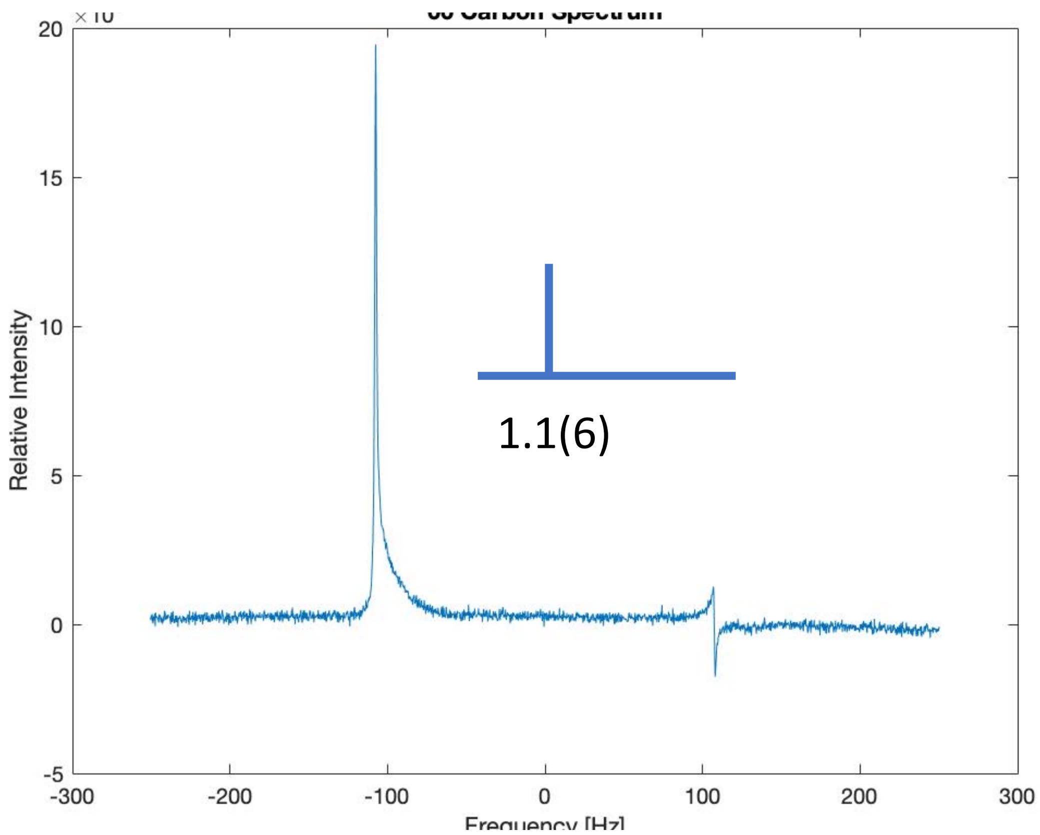

- A $R_x\left(\frac{\pi}{2}\right)$ is applied to bring the spin into the transverse plane and the magnetic moment is measured for some time.

# Spectra & Peak Integrals of $|00\rangle$ (scaled by $10^5$)

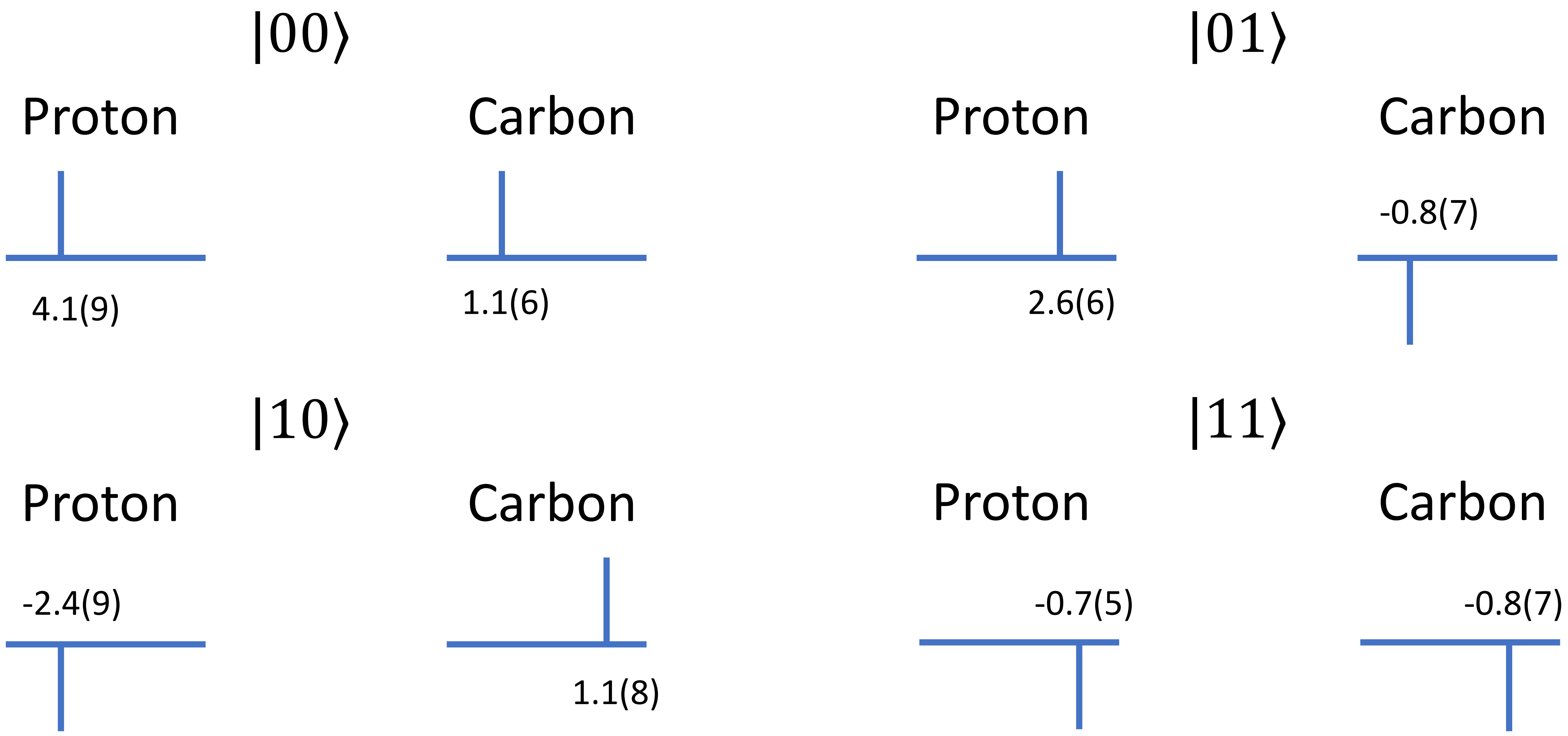
Proton




Carbon

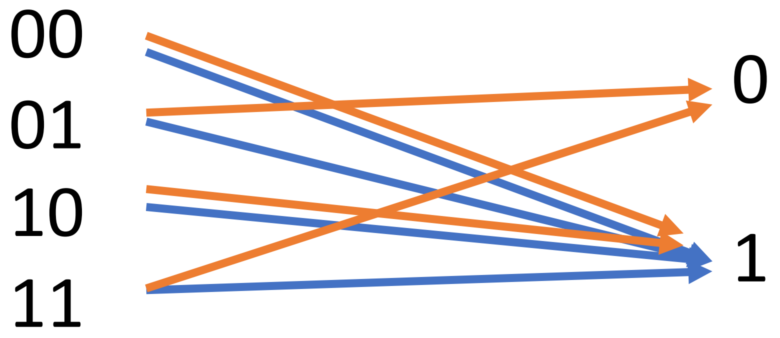


# Spectra & Peak Integrals of Eigenstates
(scaled by $10^5$)

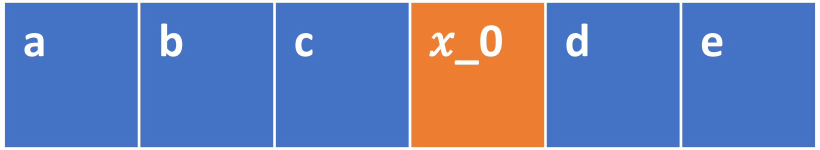


# Two “Fast” Quantum Algorithms

1. Deutsch-Jozsa Algorithm
    - Determine if $f$ is constant or faithful
    - $O(2^n)$ on classical computer
    - 1 query is sufficient on quantum computer

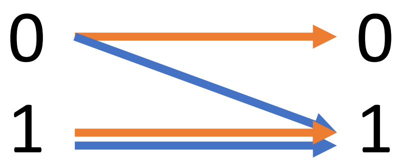



2. Grover’s Algorithm
    - Search for an unknown variable $x_0$
    - $O(N)$ on classical computer
    - $O(\sqrt{N})$ on quantum computer



# Deutsch-Jozsa Algorithm on two qubits

- Finding out if a coin is fair or rigged




- Classically we need two checks:
  - Check head (evaluate $f(0)$)
  - Check tail (evaluate $f(1)$)
  - Fair coin if $f(0) \neq f(1)$, rigged otherwise

2 Queries

- On quantum computer, we can check the "middle" side:
  - Evaluate $U_f \frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  - Fair coin if $U_f|+\rangle = |00\rangle$ rigged if $U_f|+\rangle = |10\rangle$

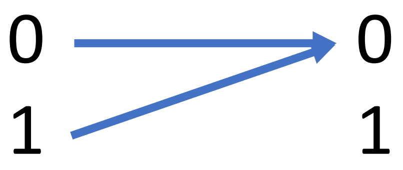
1 Queries

# Deutsch-Jozsa Algorithm Results

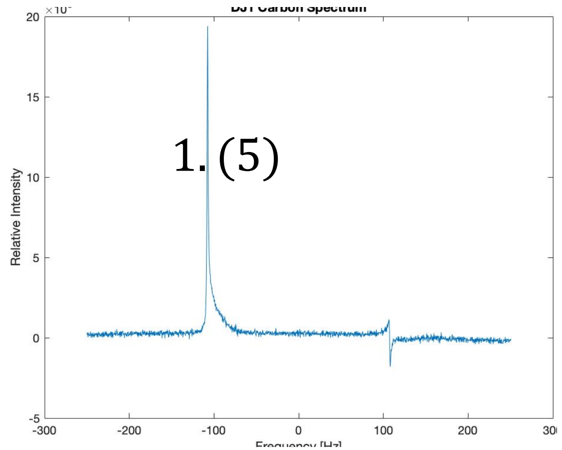
Input $f$

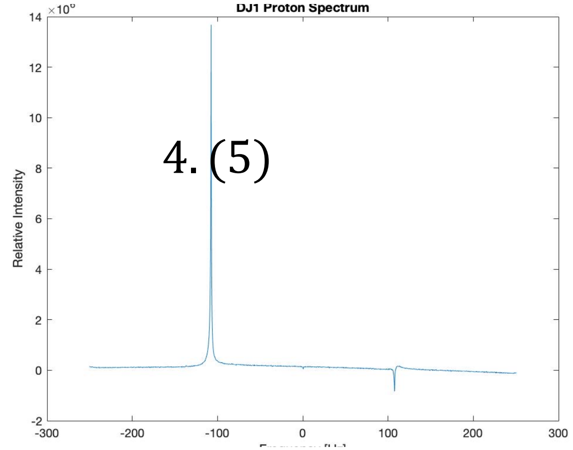
Observed Spectra

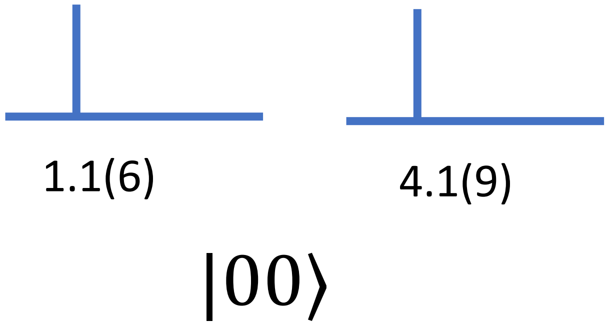
Expected Spectra




Constant

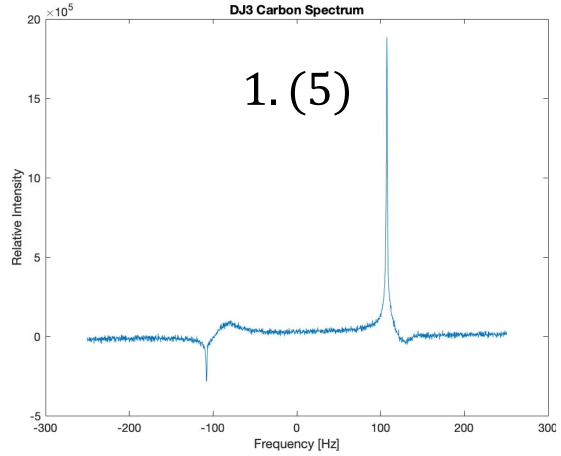


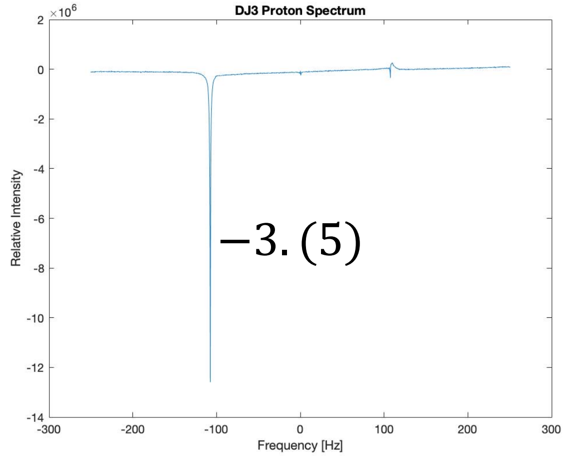



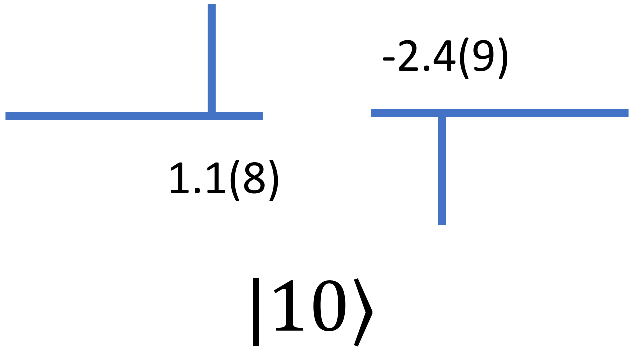





Faithful



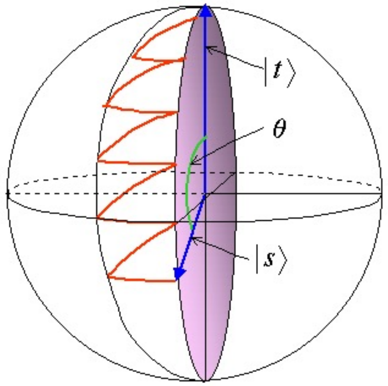
# Grover's Algorithm on two qubits

- Given $f$ such that $f(x) = -1$ iff $x = x_0$; and $f(x) = 1$ otherwise.

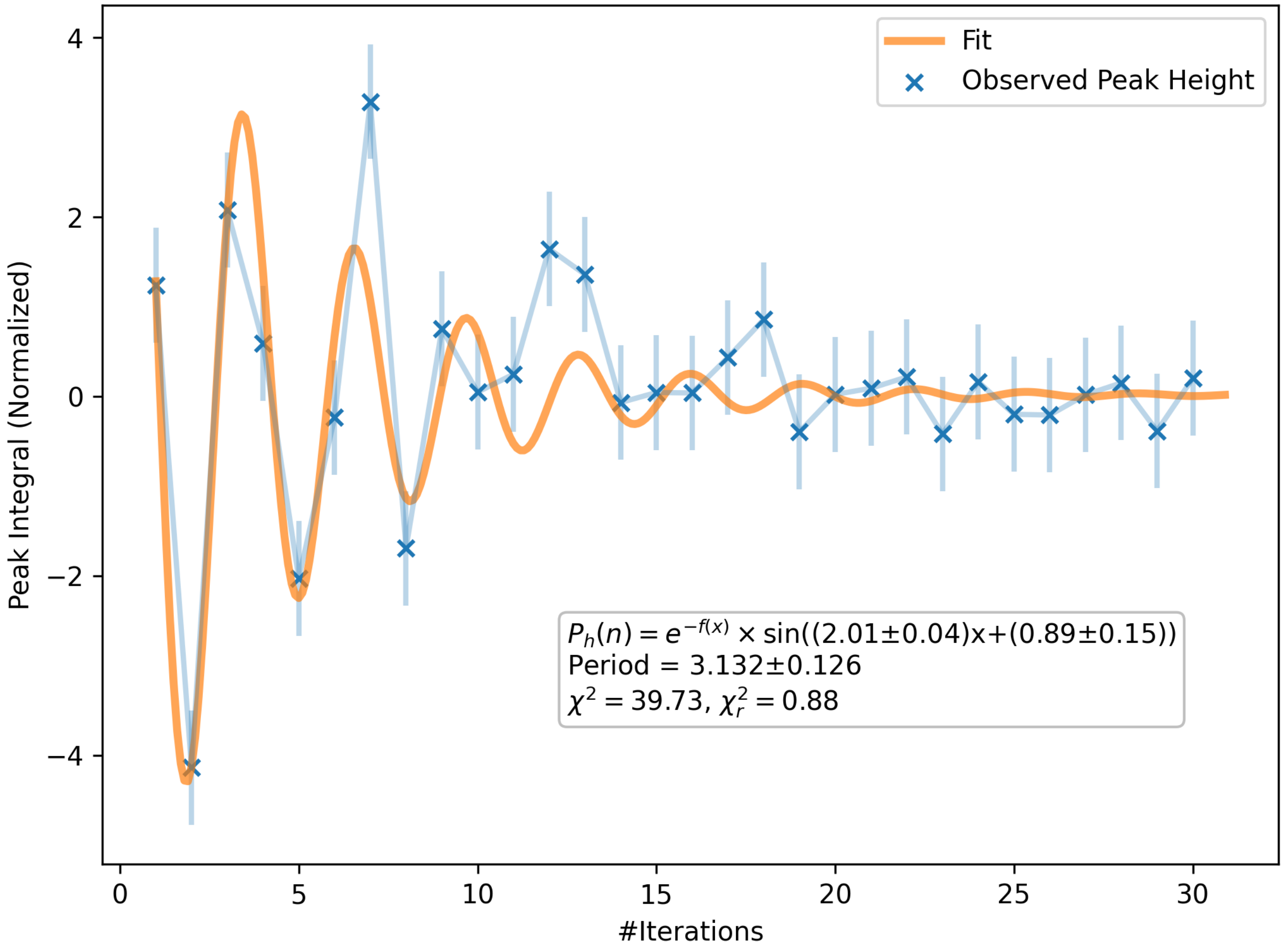
- Classically we need $O(N)$ checks
  - Worst case: $N - 1$ checks
  - Expected: $\frac{N+1}{2}$ checks

| a | b | c | d | e | f |
|---|---|---|---|---|---|
| 1 | | 1 | -1 | | 1 |

- Grover's Algorithm works by rotating a guess by $\theta = 2\arcsin\left(\frac{1}{\sqrt{N}}\right)$ each iteration towards $|x_0\rangle$
- $O(\sqrt{N})$ iterations needed total



# Search Result for $x_0 = |00\rangle$ with $H^{\otimes 2}|00\rangle$ as initial guess




Theoretical Expectation:

- Each iteration rotates our guess by $\theta = 2\arcsin\left(\frac{1}{2}\right) = \frac{\pi}{3}$.
- Recover $x_0$ after **one call**, then after every $\frac{\pi}{\theta} = \mathbf{3}$ **iterations**.

Experimental Result:

- Peak integral is large after **one iteration** → Matches with $x_0$
- Peak integral is periodic with **period $\mathbf{3.132 \pm 0.126}$** iterations
- Peak integral decays overtime

# Concluding Remarks

- We've shown quantum advantage on **query complexity**

| Algorithm | Classical Runtime | | Quantum Runtime | | Significance |
|---|---|---|---|---|---|
| Deutsch-Jozsa | 2 | $O(2^n)$ | 1 | 1 | Oracle separation of **QEP** & **P** |
| Grover | 2.5 | $O(N)$ | 1 | $O(\sqrt{N})$ | Potential Practical Speed Up |

- <u>*NOT*</u> the same as time complexity
- Future Direction: show quantum advantage for space complexity?

# Thank you!

Questions?

# Back-up Slides

# Error Analysis

- Numerical schemes: 3% (for hydrogen) and around 10% (for carbon).
  - Improper shimming → The spectrum is asymmetric.

- Uncertainty in the measurement of pulse widths propagates as the circuit grows larger.

- Background noise: Additional <1% uncertainty in the FID

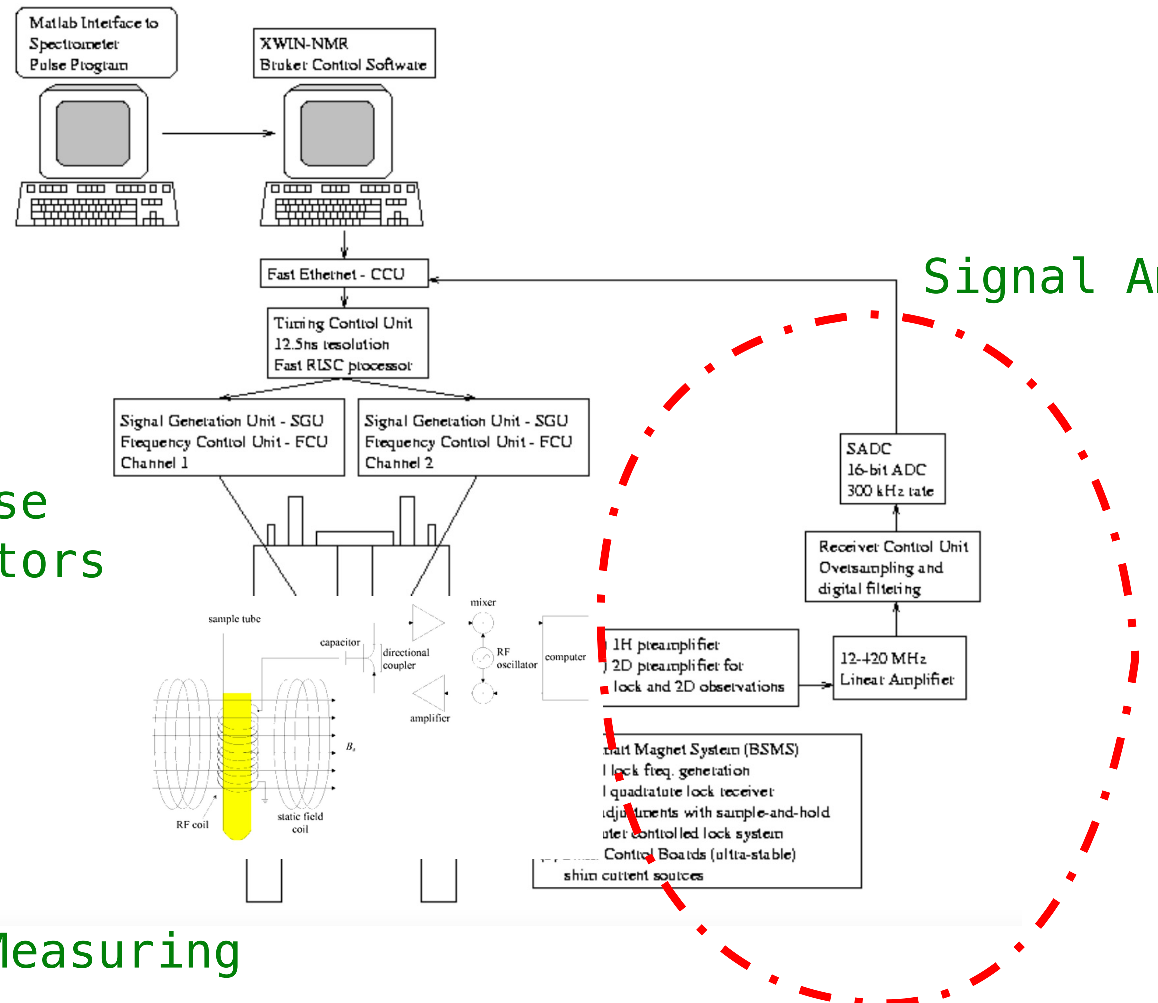
- The uncertainties are larger for the Carbon qubit
  - Faster decoherence for Carbon (Smaller $T_1, T_2$ time)
  - Higher pulse width for a 90-pulse on Carbon

# Measurement Apparatus

Control and Job Assigning

Signal Amplification

RF Pulse Generators




Measuring

# Calibrations

| Description | Measurement Value | Method/Comments |
|---|---|---|
| J, coupling constant | $215 \pm 1\ [Hz]$ | Difference between two peaks |
| $\phi_H, \phi_C$ | $[10.(5), -40.(5)] deg$ | Using $t_{90}^H$ = 10 ms, $t_{90}^C$ = 22 ms. Run NMRCalidb and rephase until imaginary part is <10% real part. |
| $t_{90}^H,\ t_{90}^C$ | $[10 \pm 1, 22 \pm 1] ms$ | Using $\phi_H, \phi_C$ as above and run NMRCalib $\Delta = 1,2, \dots, 30$ ms delay. Choose $t_{90}^H,\ t_{90}^C$ to be arg max of the total response integral |
| $T_1^H, T_1^C$ | [19.(5), 12.(5)] s | Using 90-Δ-180 for $\Delta = 1{,}500, \dots, 10000$ ms and fit exponential decay to peak integrals |
| $T_2^H, T_2^C$ | [2.(2), 1.(2)] s | Fit Lorentzian |

# Pure State Preparation

- For thermal state $\rho_{therm} = diag[a, b, c, d]$, cyclically permutating the last three canonically basis and averaging yields a new state $\rho_{avg} = diag[3a, 1-a, 1-a, 1-a]$, since $tr(\rho_{therm}) = 1$. This is effectively a pseudo pure state $|00\rangle$.
- We can apply $R_x^C(\pi)$ and $R_x^H(\pi)$ to obtain the remaining pure states.

| State | Left H Peak | Right H Peak | Left C Peak | Right C Peak |
|---|---|---|---|---|
| 00 (Id) | **4.14 + 0.71i** | 0.48 + 0.81i | **1.09 + 0.09i** | 0.09+ 0.06i |
| 01 (X_c) | 2.05 - 0.63i | **2.59 + 0.36i** | **-0.82 -0.09i** | -0.18-0.15i |
| 10 (X_h) | **-2.45-0.78i** | 0.95 -1.20i | -0.11-0.30i | **1.13-0.24i** |
| 11 (X_cX_h) | -0.09-0.35i | **-0.70-0.02i** | 0.00+0.04i | **-0.78+0.27i** |

# CNOT and near CNOT performance

| Near CNOT | Left H Peak | Right H Peak | Left C Peak | Right C Peak |
|---|---|---|---|---|
| 00→00 | **3.88+0.86i** | 0.36+0.58i | **1.13 + 0.10i** | -0.10+0.16i |
| 01→01 | 1.36-0.40i | **2.03-0.25i** | **-0.93-0.53i** | -0.02-0.23i |
| 10→11 | -1.47-0.32i | **-1.16 -0.03i** | -0.13+0.22i | **-0.93+0.66i** |
| 11→10 | -**0.81-0.95i** | -1.13-0.31i | -0.04-0.05i | **0.50-0.78i** |

| CNOT | Left H Peak | Right H Peak | Left C Peak | Right C Peak |
|---|---|---|---|---|
| 00→00 | **3.23+1.23i** | 0.33+0.60i | **0.95 -0.02i** | -0.12+0.11i |
| 01→01 | 1.64-0.11i | **1.86+0.24i** | **-0.74-0.10i** | -0.10-0.26i |
| 10→11 | -1.90-0.20i | **-1.15-0.55i** | -0.12+0.16i | **-0.87+0.60i** |
| 11→10 | **-1.62-0.97i** | -0.61+0.10i | -0.02-0.09i | **0.54-0.43i** |

# 1. Deutsch-Jozsa Algorithm Details

Classically:

- We say a function $f$ is
  - constant if $f(x) = 0$ or $f(x) = 1$ for all $x$,
  - Faithful if $f(x) = 0$ on exactly half of $x$, and $f(x) = 1$ otherwise
- Given function $f$ guaranteed to be constant or faithful, $\boldsymbol{O(2^{|x|})}$ **queries** to $f$ is needed to decide whether $f$ is constant.

Quantum Analogue:

- Define $U_f$ for a function $f$:
  $U_f|x\rangle \otimes |y\rangle = |x\rangle \otimes |y \oplus f(x)\rangle$
- Exactly **one query** to $U_f$ is sufficient:

$$R_y^H\left(-\frac{\pi}{2}\right) R^C \left(\frac{\pi}{2}\right) U_f R_y^H \left(\frac{\pi}{2}\right) R^C \left(-\frac{\pi}{2}\right)|00\rangle$$
$$= \frac{1}{2}\left[(-1)^{f(0)}(|0\rangle - |1\rangle) + (-1)^{f(0)}(|0\rangle + |1\rangle)\right] \otimes |0\rangle$$

- Which is $\pm|00\rangle$ if $f$ is constant, and $\pm|10\rangle$ otherwise

# 1. Deutsch-Jozsa Algorithm Details

- When $|x| = 1$, there are a total of 4 different functions:

| | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ | $f_4(x)$ |
|---|---|---|---|---|
| Input 0 | 0 | 1 | 0 | 1 |
| Input 1 | 0 | 1 | 1 | 0 |
| Type | Constant | | Faithful | |
| $U_f$ | $I$ | $R_x^C(\pi)$ | $CNOT$ | $R_x^C(\pi)CNOT$ |

- Running $R_y^H\left(-\frac{\pi}{2}\right)R^C\left(\frac{\pi}{2}\right)U_f R_y^H\left(\frac{\pi}{2}\right)R^C\left(-\frac{\pi}{2}\right)|00\rangle$ yields the following output:

# Constant
# $|00\rangle$

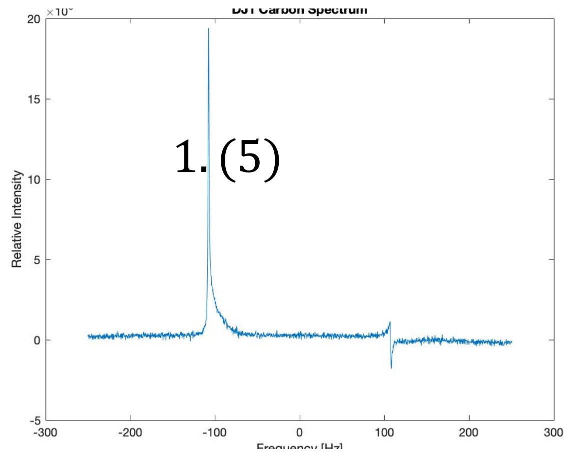
Carbon

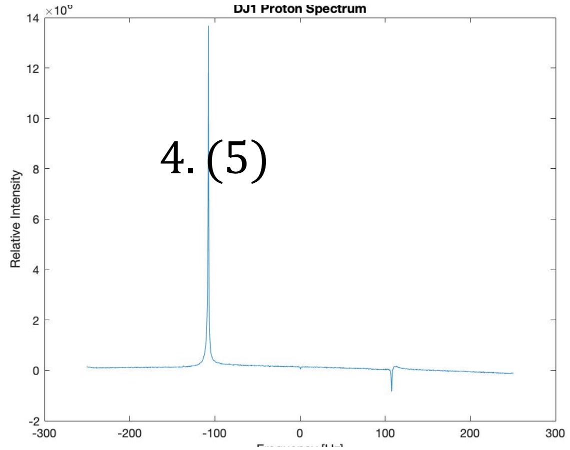
1.1(6)

Proton

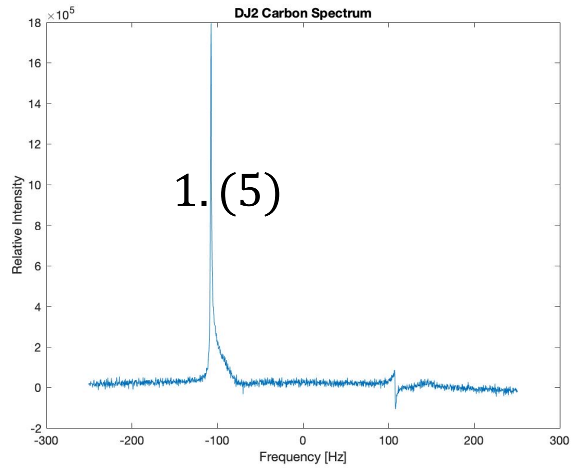
4.1(9)

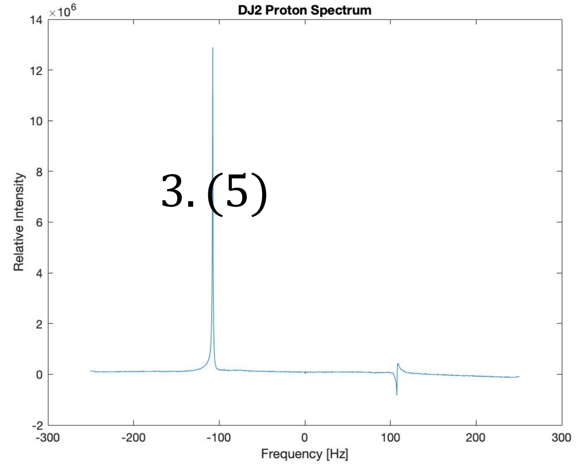
$U_{f_1}$




$U_{f_2}$



# Faithful $|10\rangle$

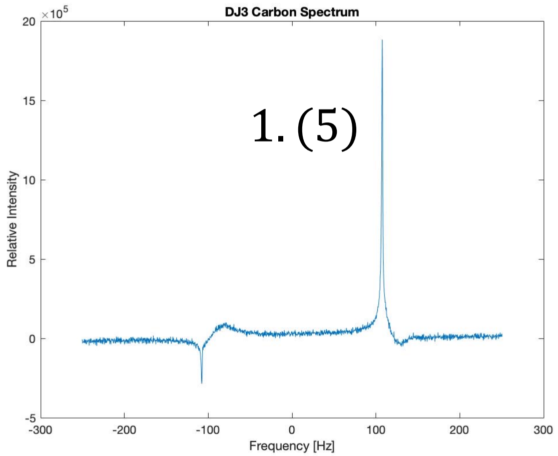
Carbon

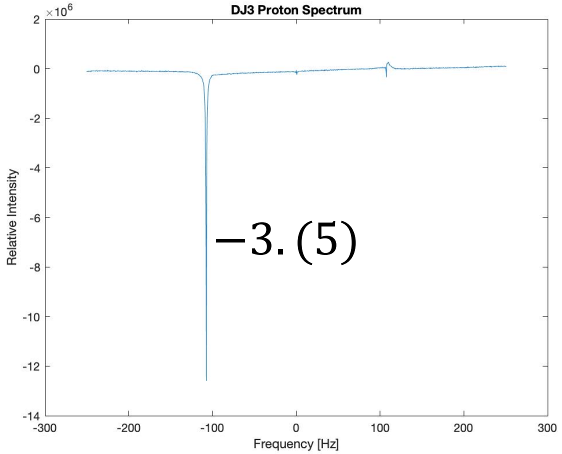
1.1(8)

Proton

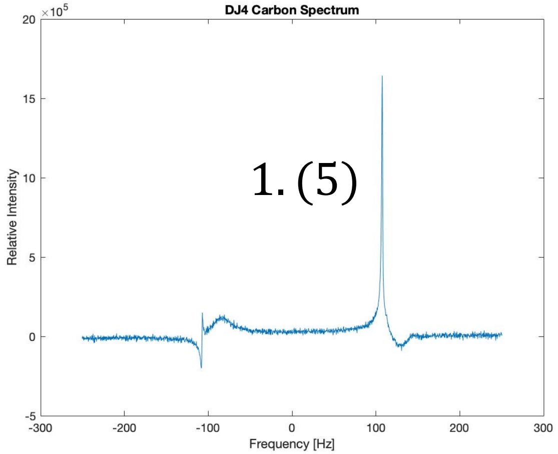
-2.4(9)

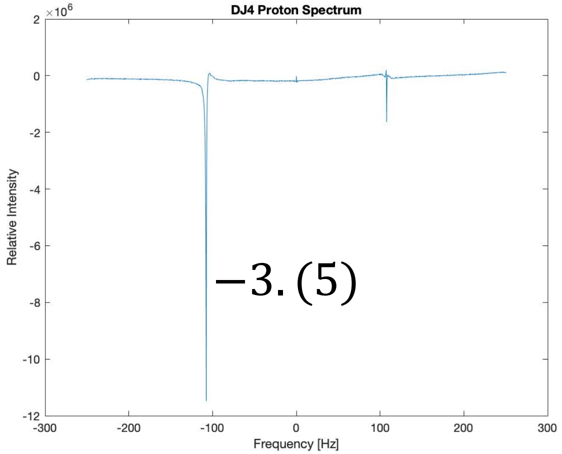
$U_{f_3}$




$U_{f_4}$



# 2. Grover's Search Algorithm Details

Classically:

- Given function $f$, where $f(x_0) = 1$ for exactly one input $x_0$, and we wish to search for $x_0$.
- Need to look through all inputs in $O(N)$ time to find $x_0$.

Quantum Analogue:

- Define $U_f$ for a function $f$: $U_f|x\rangle = (-1)^{f(x)}|x\rangle$
- Recovers $x_0$ with $O\left(\sqrt{N}\right)$ time!
- Each iteration rotates an initial guess by $\theta = 2\arcsin(\frac{1}{\sqrt{N}})$ towards $x_0$.

# Compiling Quantum Circuits –Elementary Gates

- We wrote custom class to hold quantum gates, and defined the (non-communtative) ways two operators are combined.
- We verified with qiskit that these circuit identities indeed hold.

```
%Rotation for Hydrogen
R90x_h = Gate(1, "x", 0, "x", 0);
R90nx_h = Gate(1, "-x", 0, "x", 0);
R90y_h = Gate(1, "y", 0, "x", 0);
R90ny_h = Gate(1, "-y", 0, "x", 0);

%Rotation for Carbon on 90 deg around x
R90x_c = Gate(0, "x", 1, "x", 0);
R90nx_c = Gate(0, "x", 1, "-x", 0);
R90y_c = Gate(0, "x", 1, "y", 0);
R90ny_c = Gate(0, "x", 1, "-y", 0);

%Hadamard Gate
H_c = R90y_c + R90x_c + R90x_c
H_h = R90y_h + R90x_h + R90x_h
H = H_c + H_h

%Phase Shift
P = wait + R90ny_h + R90nx_h +R90y_h+ R90ny_c + R90nx_c+R90y_c

%Wait Operator:
wait = Gate(0, "x", 0, "x", 1000/2/215);

%Near CNOT Gate
rCNOT =  R90x_c + wait + R90ny_c;

%CNOT Gate
CNOT =  R90nx_h + R90y_h + R90x_h+ R90x_c + R90y_c+ wait + R90ny_c;

%Empty (Identity) Gate
GE = Gate(0,"x",0,"x",0)
```

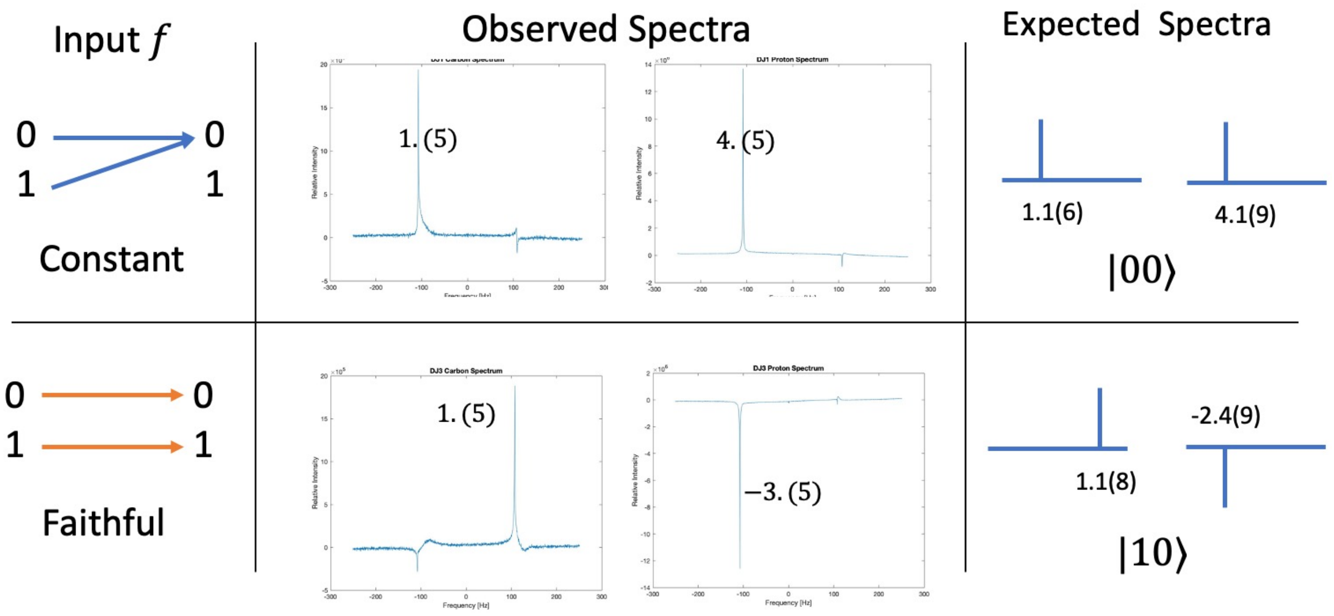
# Compiling Quantum Circuits –Algorithms

```
% DJ Functions
U1 = GE
U2 = R90x_h + R90x_h
U3 = CNOT
U4 = U3 + U2

dj1 = R90ny_c+R90y_h
dj2 = R90y_c+R90ny_h
```

```
%Grover Oracles
O11 = wait + R90ny_h+R90x_h+R90y_h+R90ny_c+R90x_c+R90y_c
O00 = wait + R90ny_h+R90nx_h+R90ny_h+R90ny_c+R90nx_c+R90ny_c
O10 = wait + R90ny_h+R90nx_h+R90ny_h+R90ny_c+R90x_c+R90y_c
O01 = wait + R90ny_h+R90x_h+R90y_h+R90ny_c+R90nx_c+R90ny_c

G00 = O00 + H + P + H
G01 = O01 + H + P + H
G10 = O10 + H + P + H
G11 = O11 + H + P + H
```

# Thumbnail

## Deutsch-Jozsa Output




## Grover Output