FDD3280 – Quantum Computing for Computer Scientists

Amplitude Amplification

Stefano Markidis

Intended Learning Outcomes

- Describe and implement the amplitude amplification and Grover iteration
- Get a geometrical intuition about amplitude amplification.

Phase Manipulation Primitive

- We introduce the first *quantum primitive* allowing us to manipulate superpositions into a form that we can **reliably READ**.
- We've already noted that there are many such primitives, each suited to different kinds of problems.
    - The first we will cover is ***amplitude amplification***.

Amplitude Amplification

$\theta \to |\ |^2$

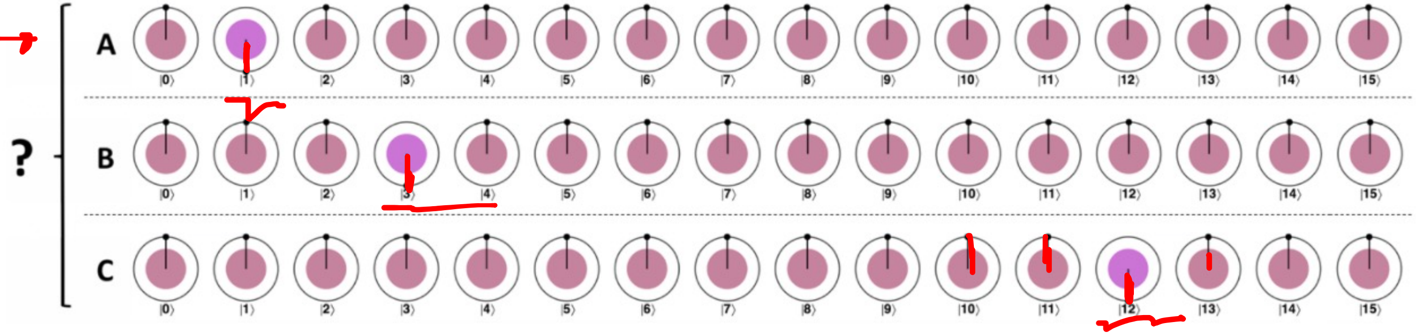
Meas

- Very simply, amplitude amplification is a tool that converts **inaccessible *phase differences*** inside a QPU register into **READable *magnitude* differences** (and vice versa).
- Given that amplitude amplification converts phase differences into magnitudes, you might think that *magnitude* amplification would be a better name.
  - However, "**amplitude amplification**" is commonly used in the wider literature to refer to the kind of primitive we describe here.

Motivational Example

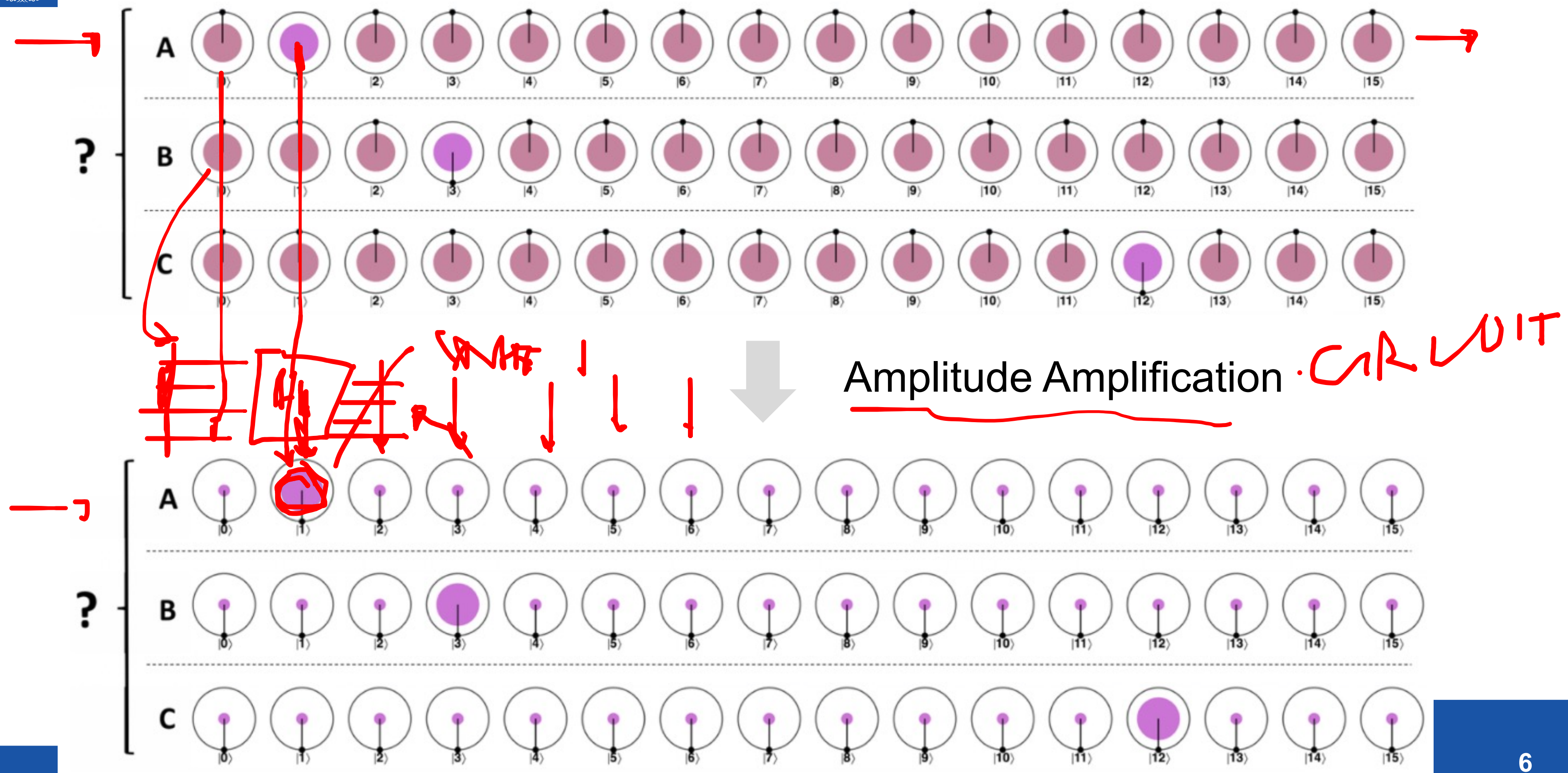
$2^4 - 0 - 15$

- For example, suppose we have a four-qubit QPU register containing one of the three quantum states (A, B, or C), but we don't know which one.



- These three states have a different certain value phase-flipped.
  - We'll call that the ***marked value***.
- Reading a QPU register will return an evenly distributed random number
  - At the same time, such READs will destroy the phase information in the register.

Amplitude Amplification Effect

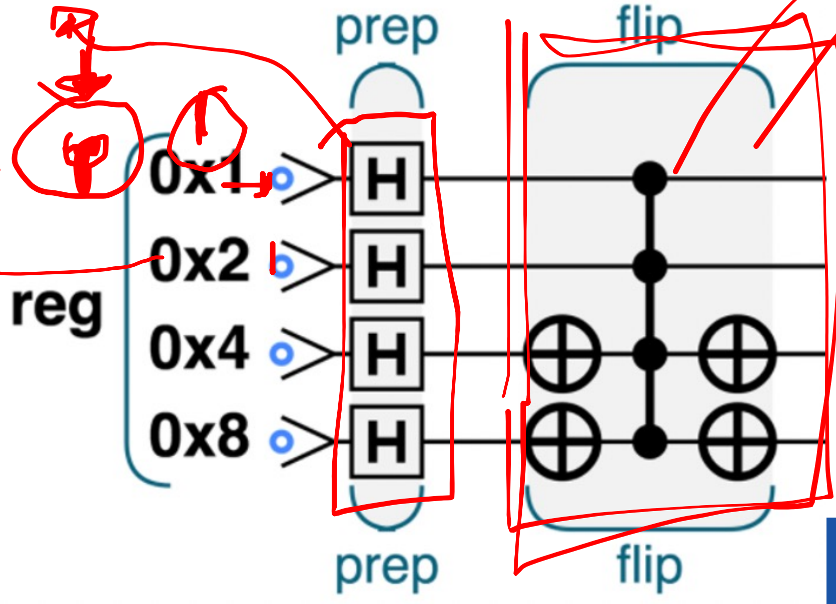


Marking the Qubit (Flip Primitive)

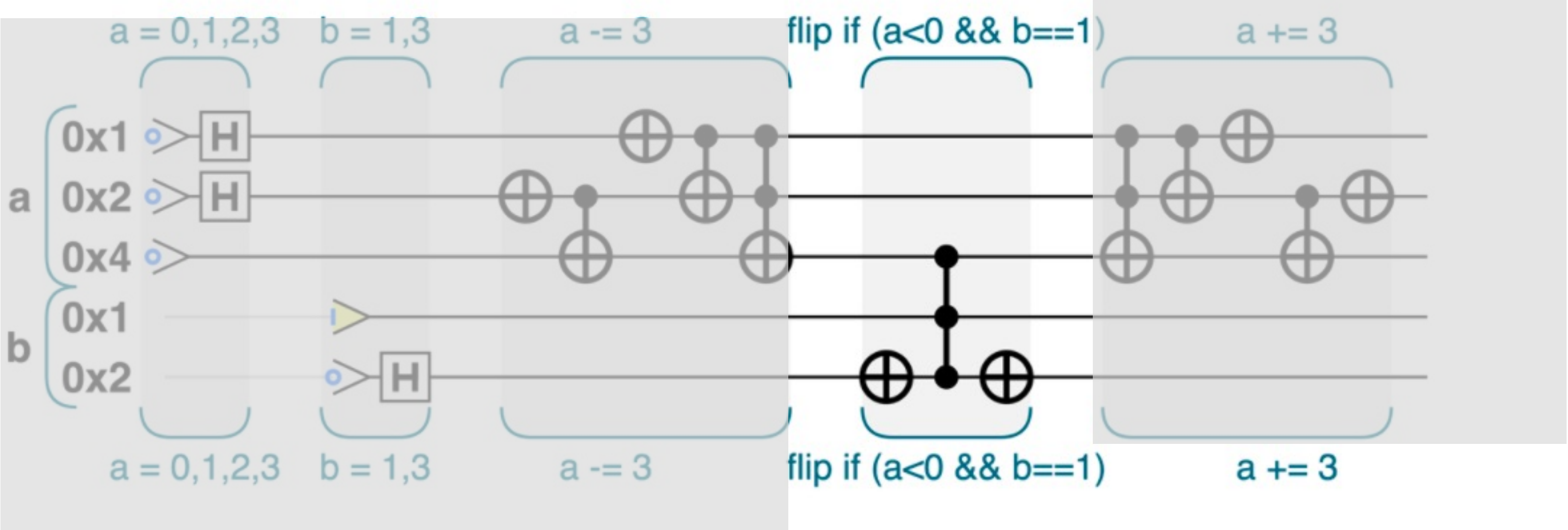
To mark the qubit (flip), we use takes our register initially in state |0⟩ and *marks* one of its values.

Put qubits
In
Superposition

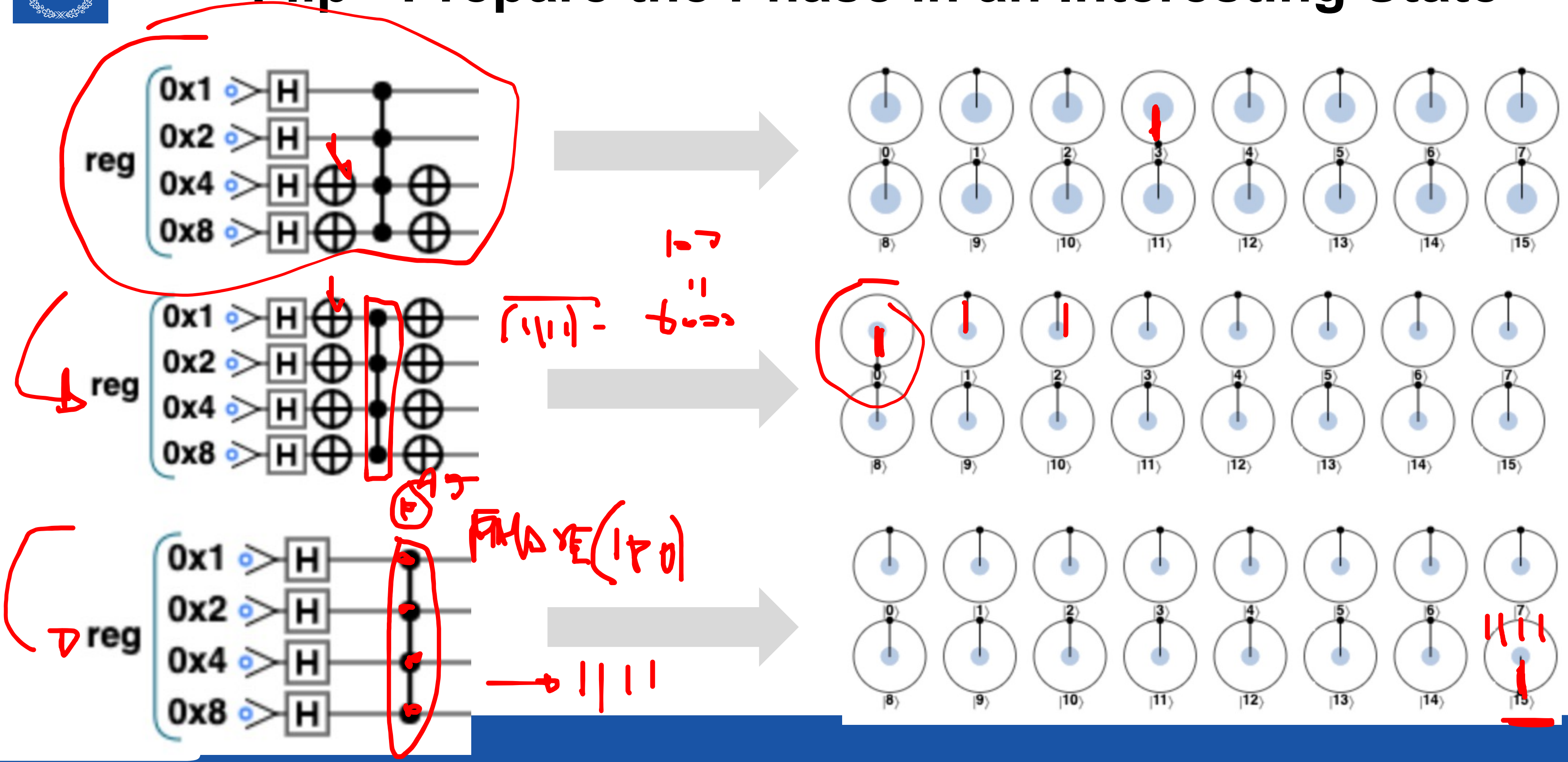
Mark
A
Qubit



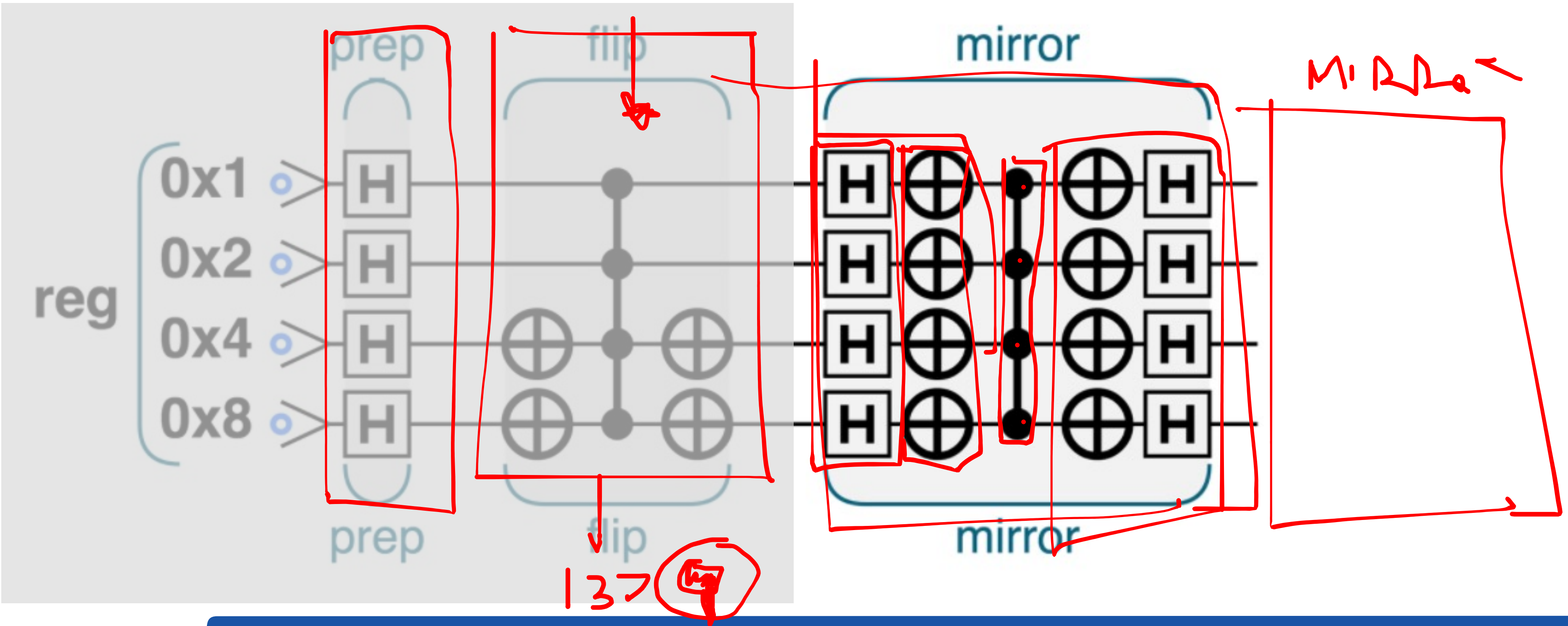
Mark a Qubit – Conditional Quantum Logic



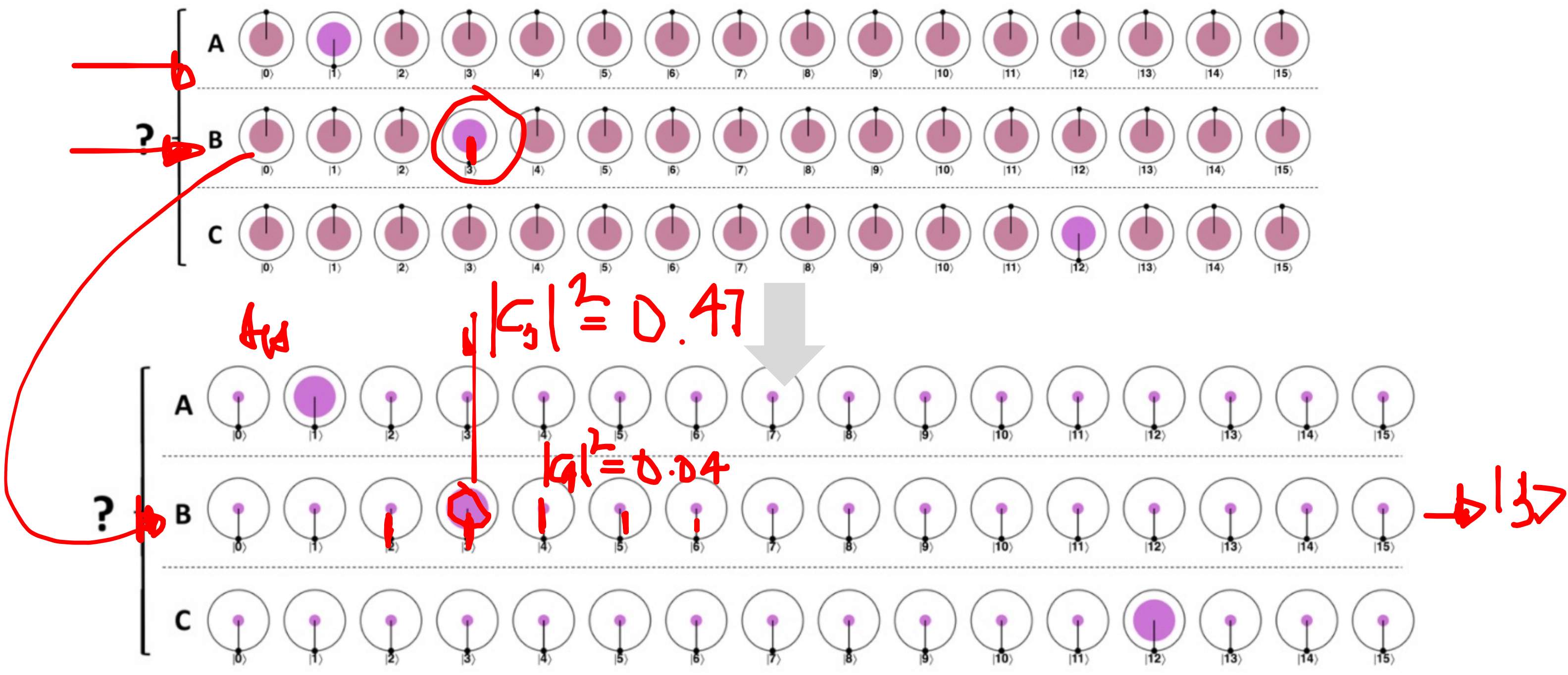
Flip - Prepare the Phase in an Interesting State



Mirror to Measure the Phase



Amplitude Amplification



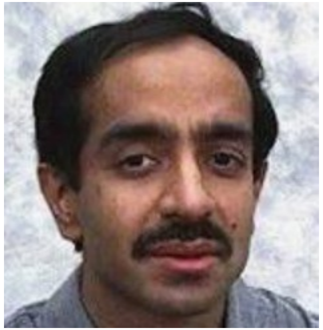
The **magnitudes within each state** are now very different, and **performing a READ on the QPU register is very likely** (though not certain) to reveal which **value had its phase flipped**,

Probability of Finding the Marked Value

- Instead of all values having the same probability of 6.25% (all the states same probability) the marked value now has a READ probability of about **47.3%**
  - with the non-marked values being at about **3.5%.**
  - At this point, READing the register gives us an almost 50% chance of obtaining the value that had its phase flipped.
- Notice the **marked *phase* is now the same as the rest of the register.**
  - In a sense, mirror has ***converted*** **the phase difference** into a **magnitude difference**.

The Mirror Operation = Grover Iteration

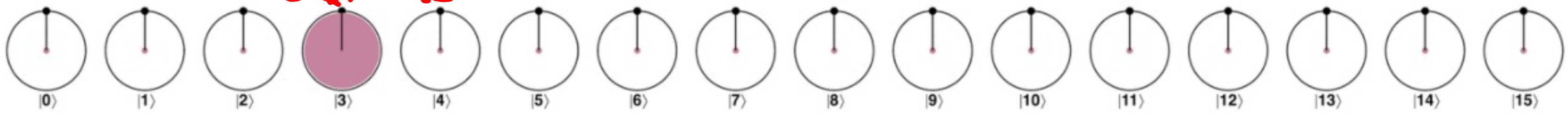
- The mirror operation is commonly called the "**Grover iteration**" in the quantum computing literature.
- **Grover's algorithm for an unstructured database search** was the first algorithm implementing the flip-mirror routine
  - The amplitude amplification primitive is a generalization of Grover's original algorithm.



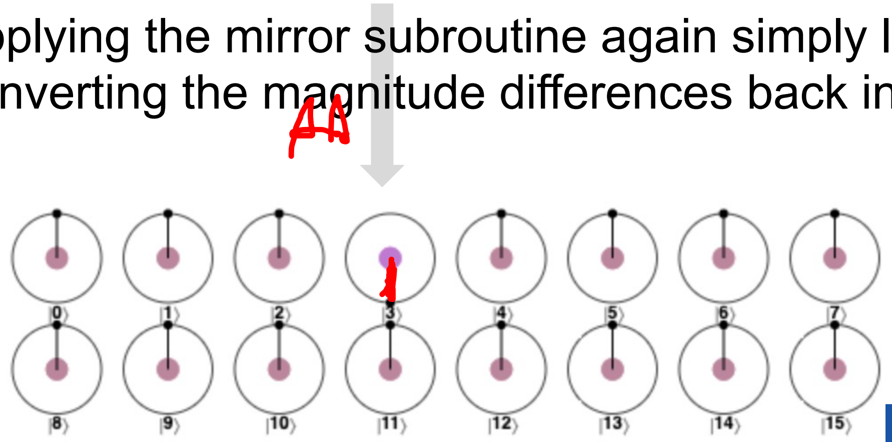
Lov Grover

Repeat Grover Iterations

Can we repeat the operation again to try to further improve our probability of success? Suppose we have the B state (i.e., flip acted on the |3⟩ value).

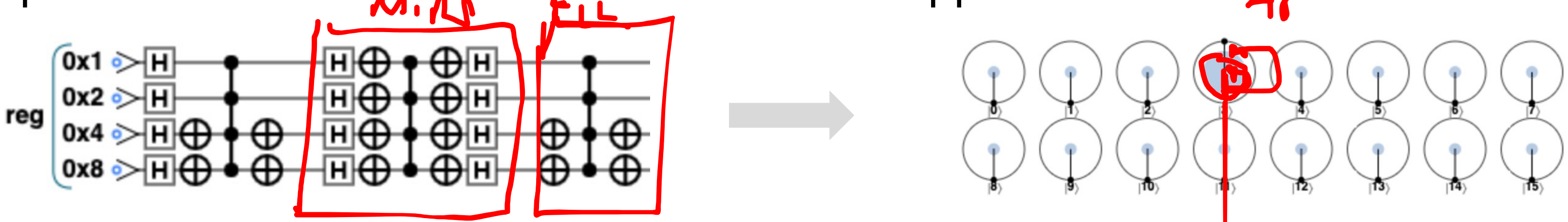


Applying the mirror subroutine again simply leaves us where we started, converting the magnitude differences back into differences of phase.

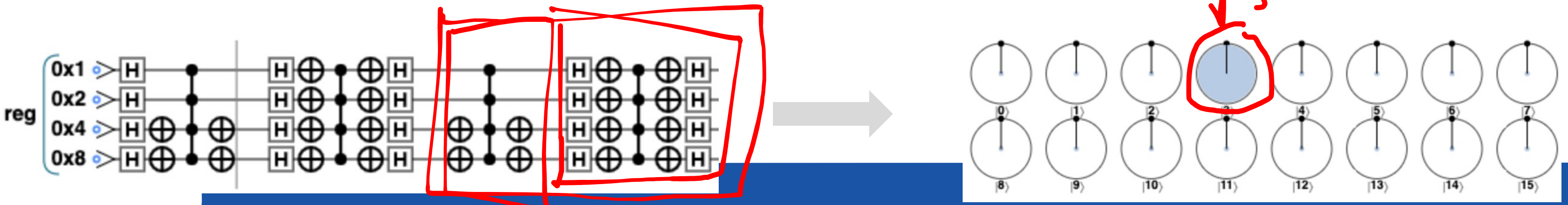


Repeat the Flip + Grover

- However, suppose that before reapplying mirror we also reapply the flip subroutine (to re-flip the marked value). This starts us out with another phase difference before our second mirror application.

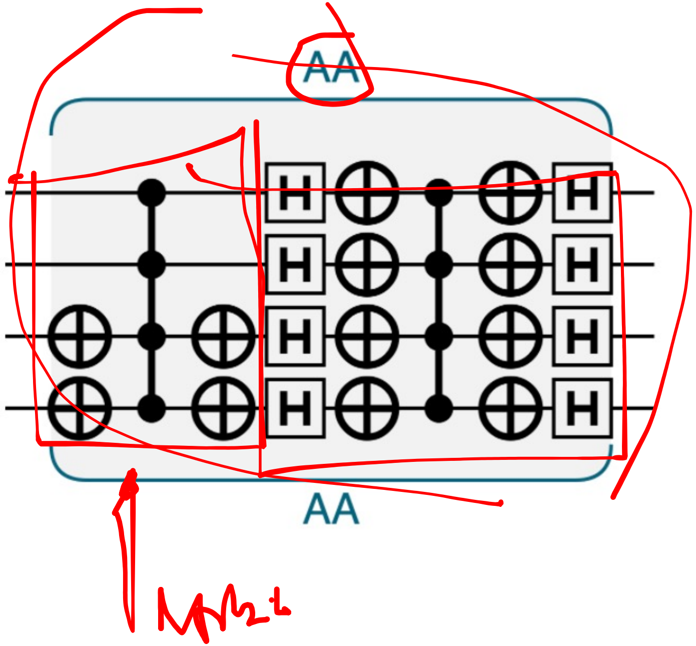


- Following two applications of flip-mirror, the probability of finding our marked value has jumped from 47.3% to **90.8%!**



The Amplitude Amplification Iteration

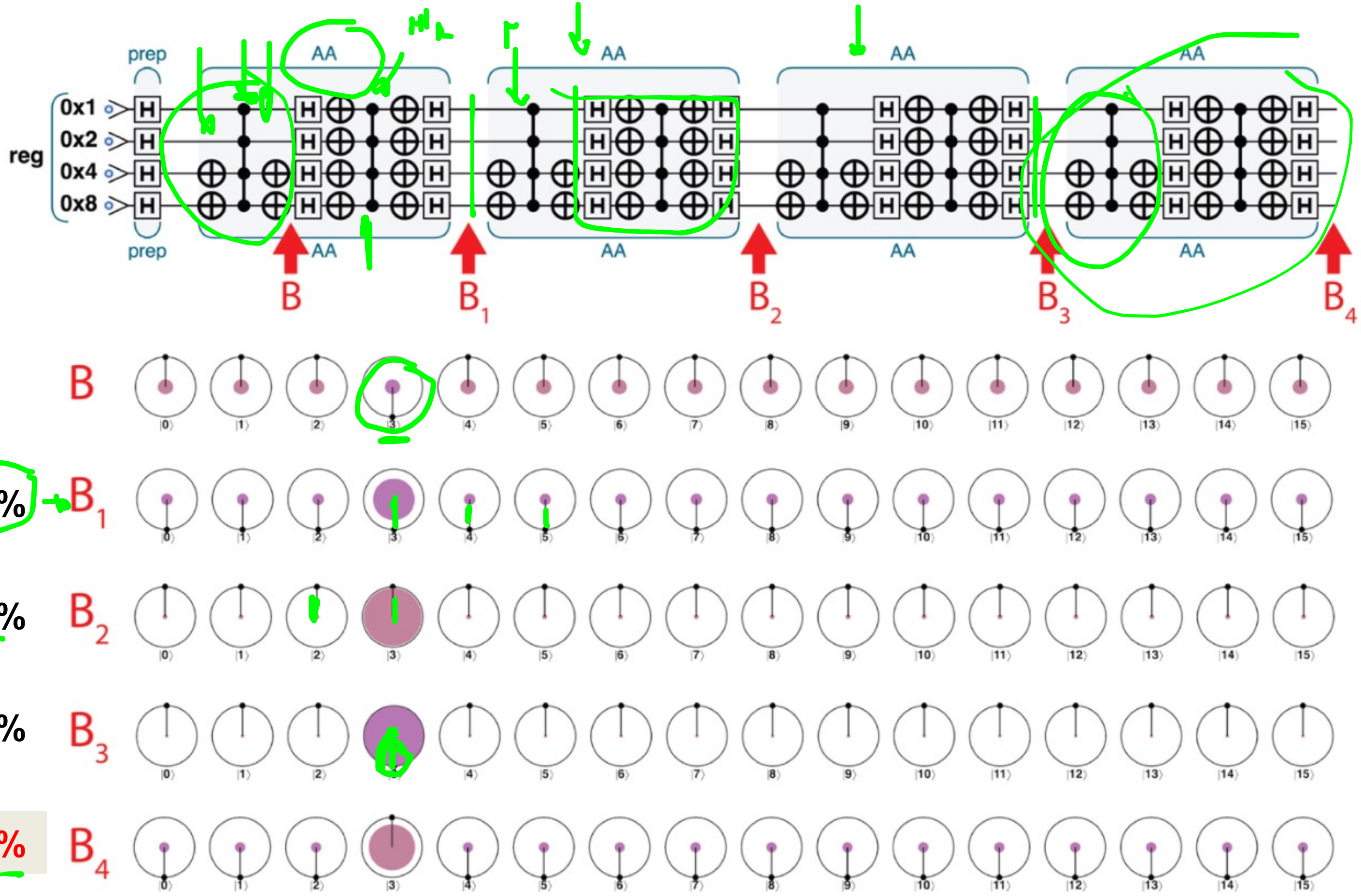
- Together, the **flip** and **mirror** subroutines are a powerful combination.
  - Flip allows us to target a value of the register and distinguish its phase from the others.
  - Mirror then converts this phase difference into a magnitude difference.



Hardwired Marking a Phase

- The AA operation assumes that we know which value we want to amplify—it's hardwired into which value the flip subroutine affects.
  - if we already know which values we should amplify, why do we need to find them?
- We've used the **flip subroutine** as the **simplest possible example**
- In a real application, **flip would be replaced** with some more **complex subroutine performing a combination of phase flips** representing logic specific to that application
- The key point is that compounding more complex phase-altering subroutines with **mirror still converts phase alterations into magnitude differences.**

Applying More Groover Iterations



Note that it is out of phase

47.3%

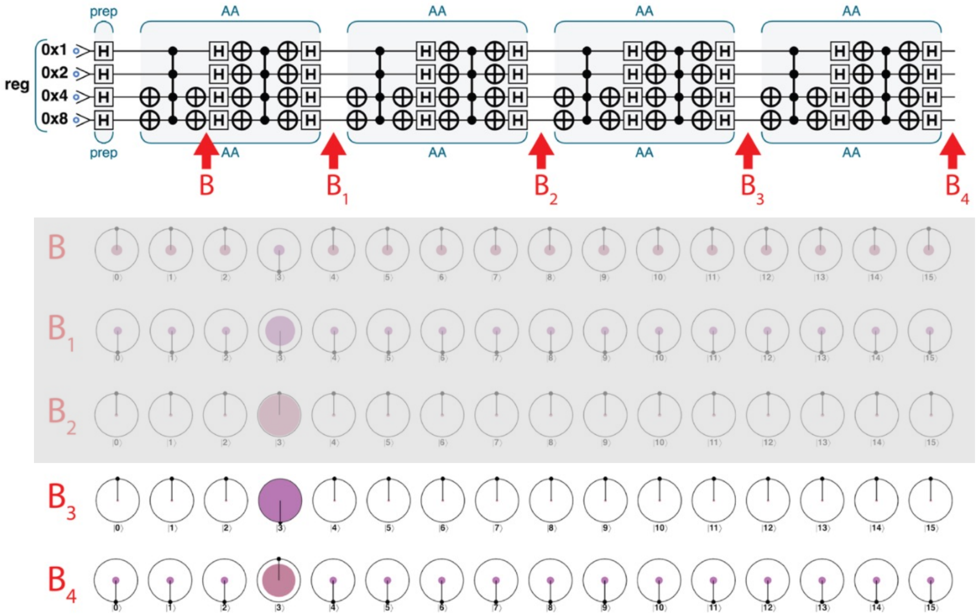
90.8%

96.1%

58.2%

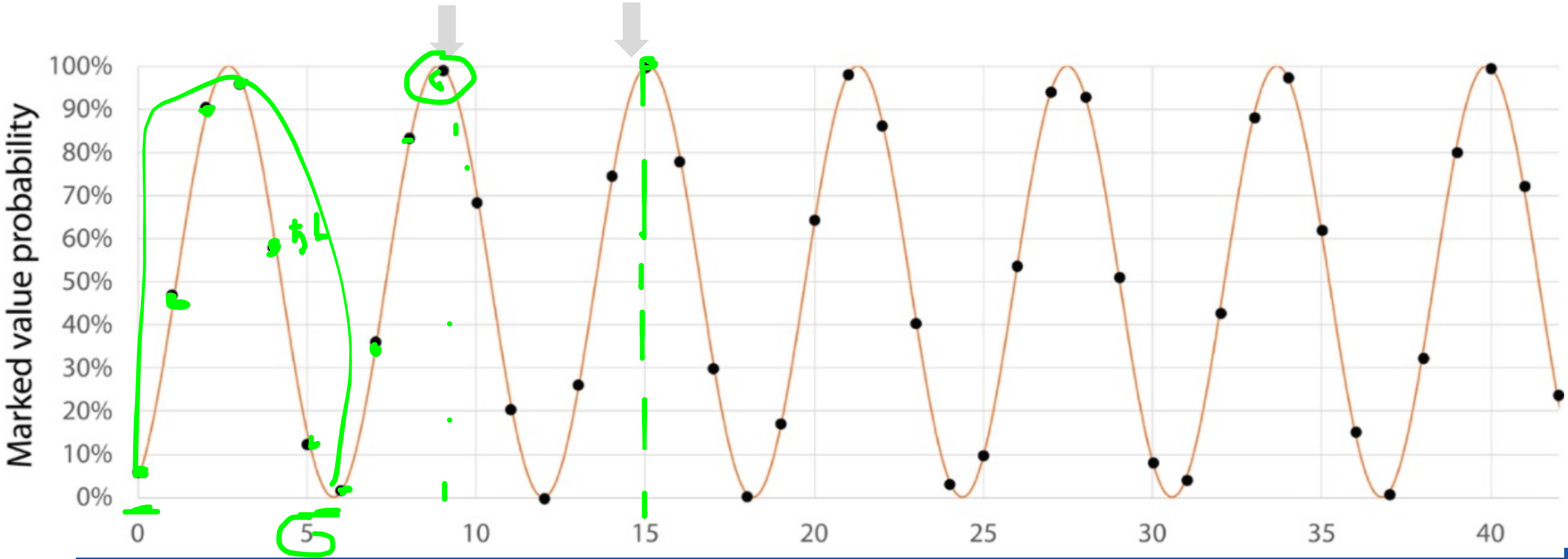
No Phase Difference from the Flip

- After B3, we will **have a magnitude difference but no phase difference**, so further AA iterations will start **diminishing magnitude differences until we end up with the original state**.
- By the time we get to B_4 our chances of successfully reading out the marked state are way down to 58.2%, and they'll **continue to drop if we apply more AA iterations**.



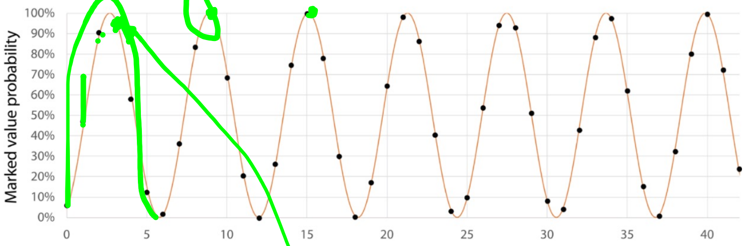
How Many Iterations?

- The plot in shows that as we continually loop through our iterations, the **probability of reading the marked value oscillates** in a predictable way.
- At the 9th or 15th iteration, we get a probability of finding the marked value of **99.9563%.**



How Many Iterations? II

- Each iteration is expensive to perform we can stop after three and attempt to harvest the answer at 96.1%.
- Even if we fail and need to repeat the entire QPU program, we'll have a 99.848% chance that one of the two attempts will succeed
- An **equation to determine the number of AA** iterations, N_{AA}, that we should perform to get the highest probability *within the first oscillation* made by the success probability (this would be the 96.1% success probability at N_{AA} = 3 in our previous example).
  - n is the number of qubits.



$$N_{AA} = \left\lfloor \frac{\pi\sqrt{2^n}}{4} \right\rfloor$$

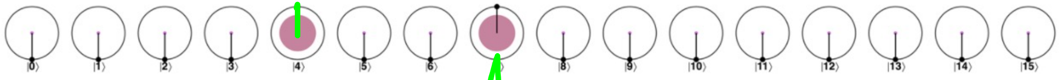
$= 3$

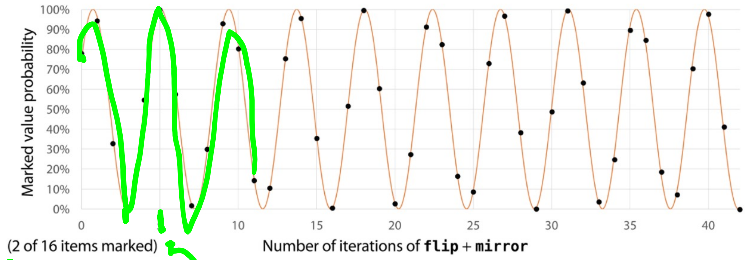
Multiple Flipped Entries

We we can try running multiple AA iterations on a register having any number of phase-flipped values.

Two Flipped Values

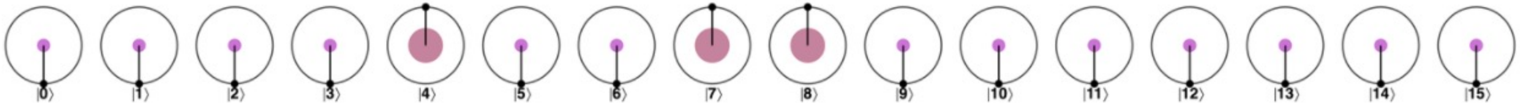
In this case we ideally want to end up with the QPU register configured so that we will READ either of the two phase-flipped values, with zero possibility of READing any others.



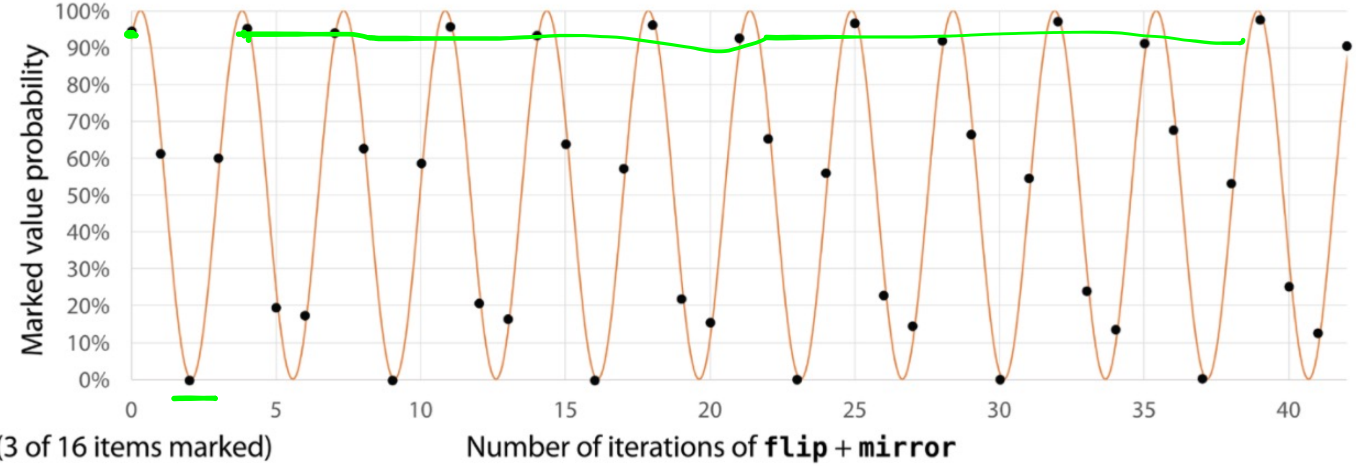


Although we still get a sinusoidally varying chance of success, we notice that the frequency of the sinusoidal wave has increased.

Three Values Flipped

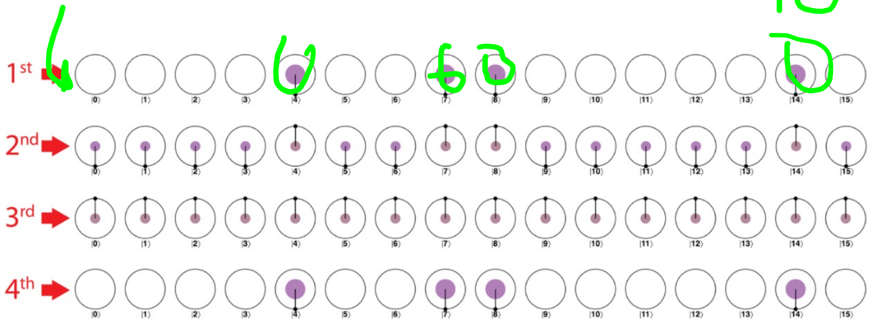


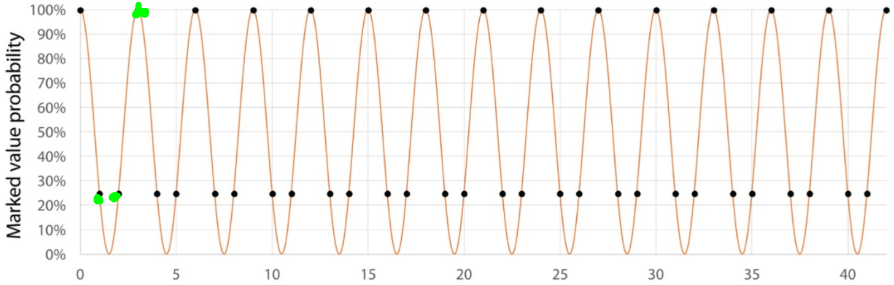
With three values flipped the wave's frequency continues to increase



Four Values Flipped

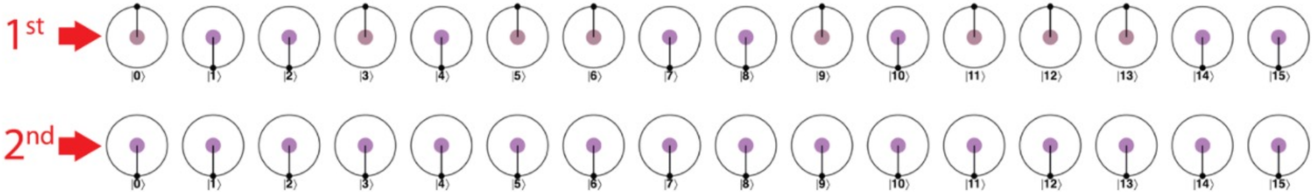
- The wave's frequency becomes such that the probability of us READing one of the **marked values repeats with every third AA** iteration that we apply.
- This ends up meaning that the very first iteration gives us 100% probability of success.

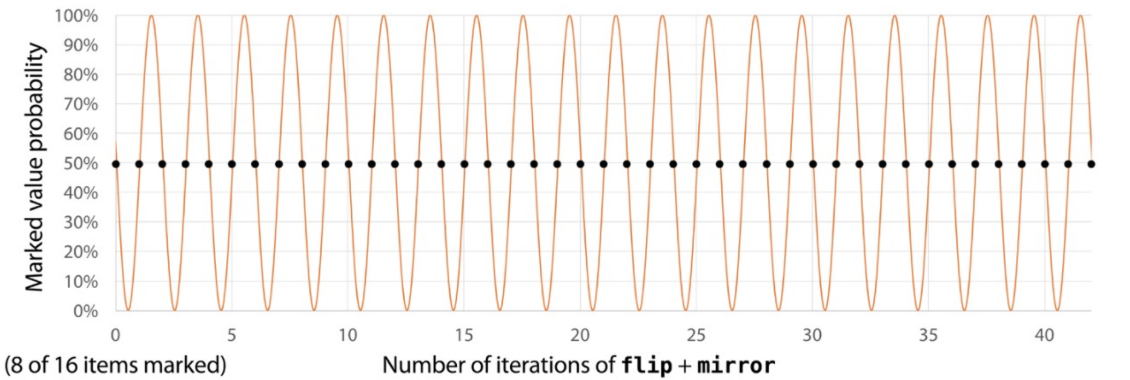




Eight Values Flipped

- Everything comes to a when we have 8 of our 16 values flipped as shown As has been mentioned in previous chapters, only the *relative* phase matters for quantum states.
- Because of this, flipping half of the values to mark them is physically the same as flipping the other half.





Number of Iterations with Multiple Markings

- It is possible to show that the frequency with which our chance of success oscillates **depends only on the *number* of flipped values**
  - **not *which* values are flipped.**
- Our equation holds for when we have multiple marked items
  - *n* is the number of qubits
  - *m* is the number of marked items

$$N_{AA} = \left\lfloor \frac{\pi}{4}\sqrt{\frac{2^n}{m}} \right\rfloor$$

How Many Iterations?

- If we *don't* know how many states are flipped, then how can we know how many AA iterations to apply to maximize our chance of success?
- When we come to employ amplitude amplification to applications in the Quantum Search algorithm, we'll revisit this question and see how other primitives can help.

Speeding Up Conventional Algorithms with AA

- AA can be used as a subroutine on many conventional algorithms, providing a quadratic performance speedup.
  - The problems that AA can be applied to are those invoking a subroutine that repeatedly checks the validity of a solution.
  - Examples of this type of problem are *Boolean satisfiability* and finding *global* and *local minima*. → OPTIMIZATION + ML
- It is in the **flip part that we *encode*** the equivalent **to the classical subroutine that checks the validity of a solution**
- the mirror part remains the same for all applications.

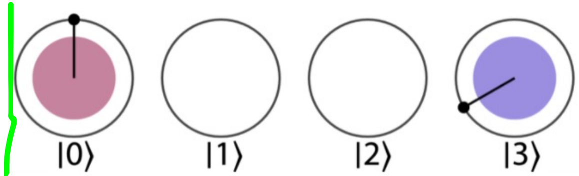
Intuition about Amplitude Amplification = mirror about average

There are two stages to amplitude amplification: flip and mirror.

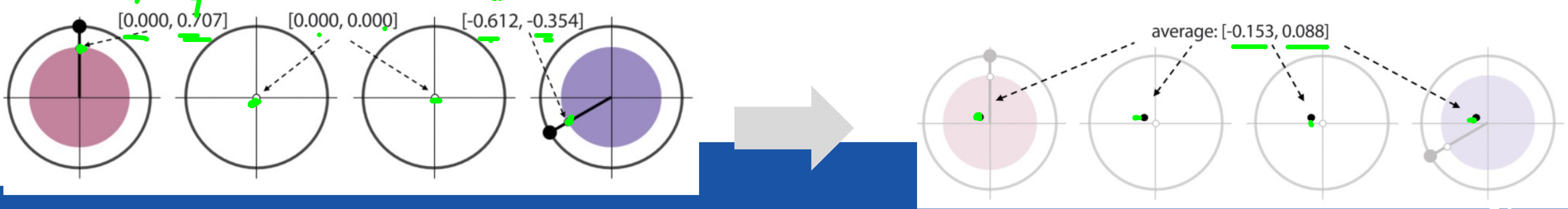
- The flip subroutine flips the phase of a marked term
- The mirror subroutine turns phase differences into contrasts in magnitude.
    - But another way to understand mirror is that it **causes each value in a state to *mirror about the average* of all values.**
        - **This explains why is called mirror**

1. Averaging

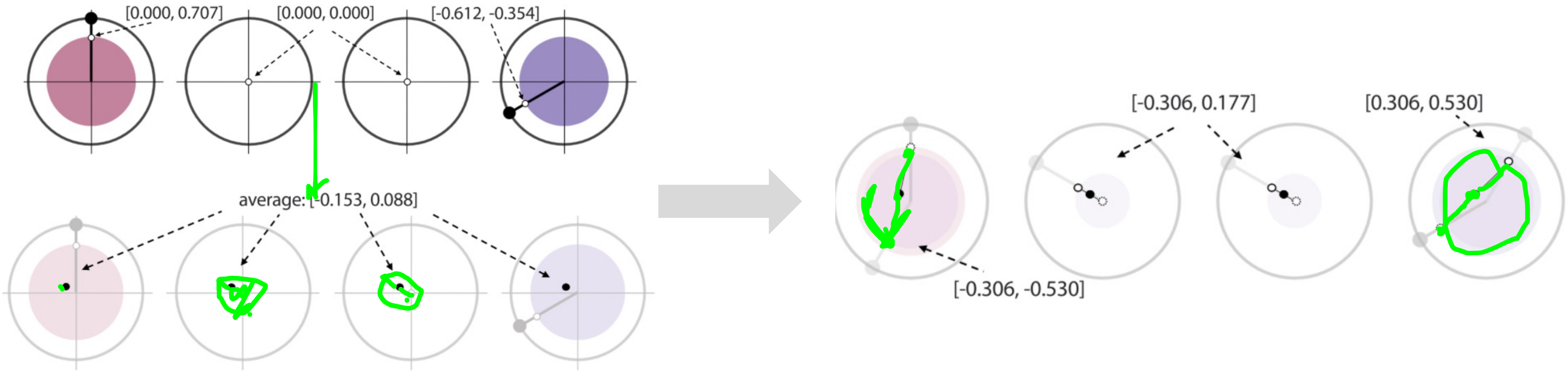
Suppose we have a two-qubit input state to the mirror subroutine, which is in a superposition of |0⟩ and |3⟩



Find the *average* of all the values (circles) → numerically averaging the *x* and *y* positions of the points within the circles (include zeros also!)



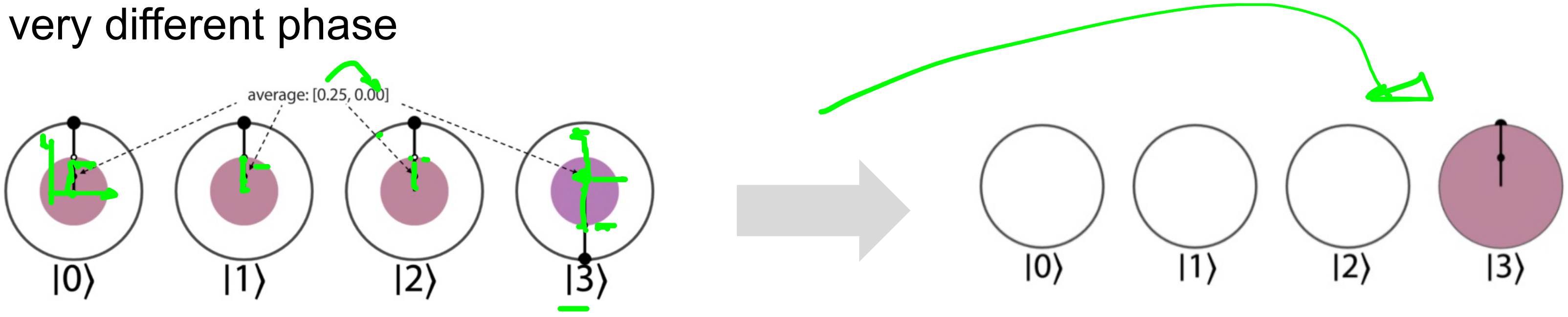
2. Flip Values About Average



- The result is that **the phase differences in our original state** have been converted into **magnitude differences**.
  - The common average of the circles is still the same.
    - This means that applying the transform again will simply return the initial state

One Marked Value

- Imagine that there are many states with similar phases, but one oddball state with a very different phase



- Given that most values are the same, the average will lie closer to the value of most of the states, and very far from the state that has the opposite phase.
  - This means that when we **mirror about the average**, the value with the different phase will "slingshot" over the average and stand out from the rest