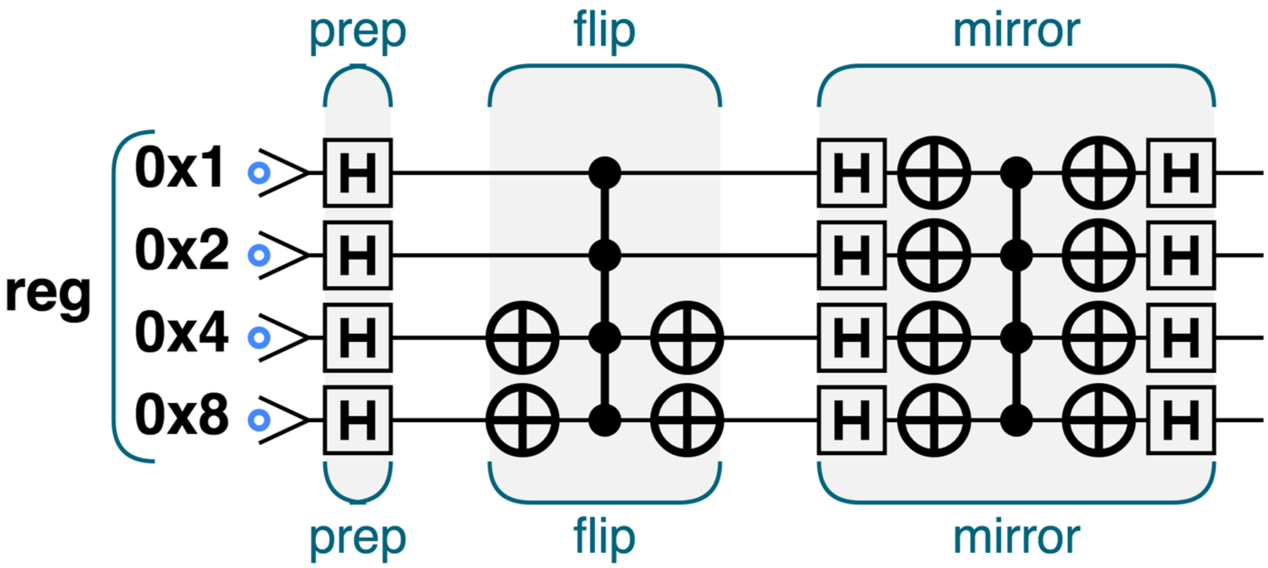
# Ch9: Amplitude Amplification

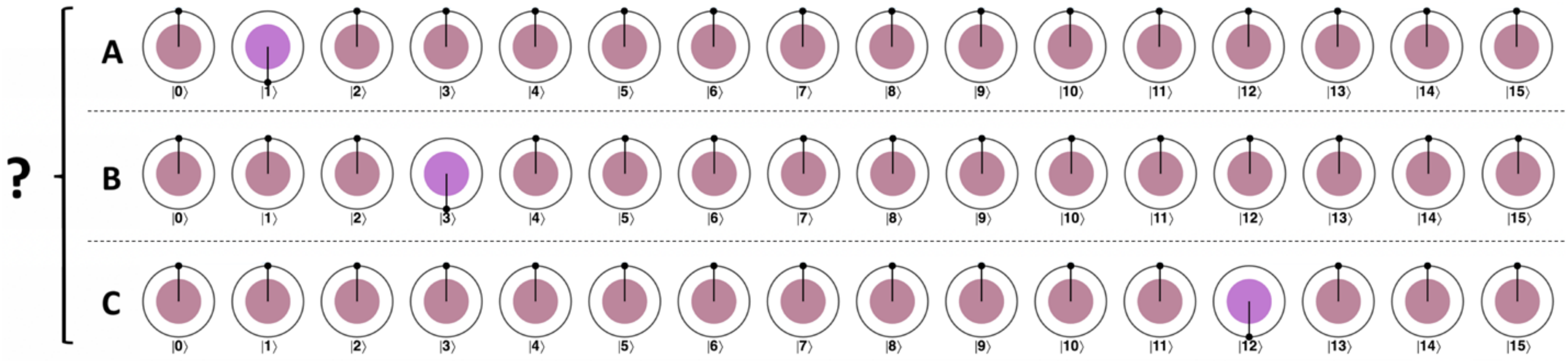
### Accessing phase information

Amplitude amplification (AA) is a tool that converts inaccessible phase differences inside a QPU register into READable magnitude differences (and vice versa). As a QPU tool, it's simple, powerful, and very useful.



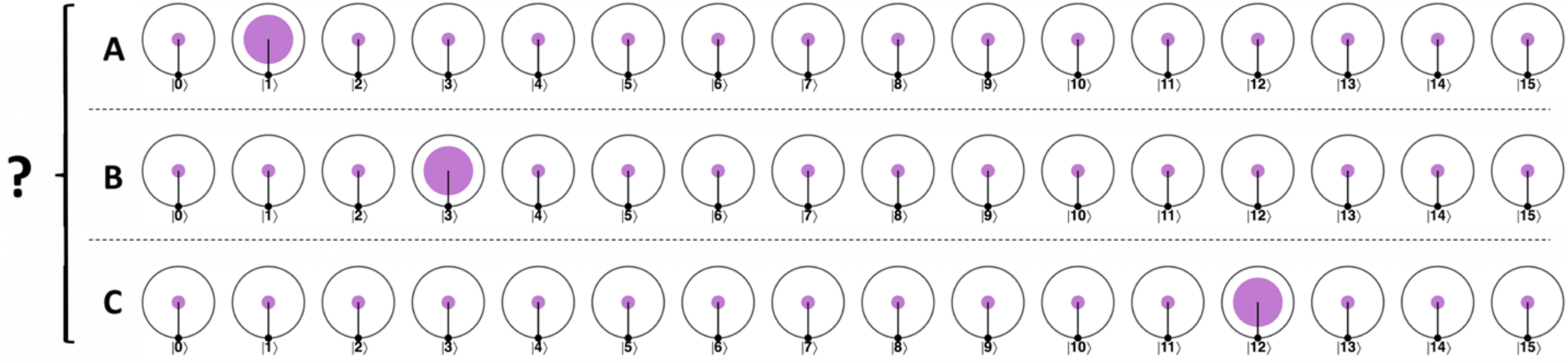
**Example applications of amplitude amplification**

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:



## Example applications of amplitude amplification

Applying the mirror subroutine to the A, B, and C states, we obtain the results:

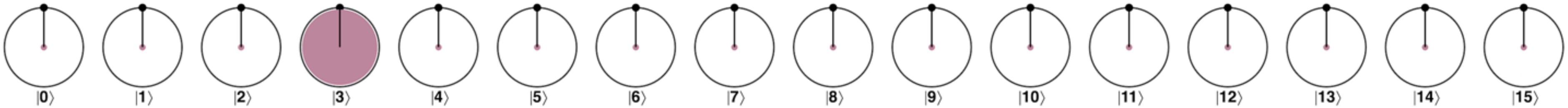


### Intuition on multiple steps

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 $|3\rangle$ value). Applying the mirror subroutine again simply leaves us where we started, converting the magnitude differences back into differences of phase.

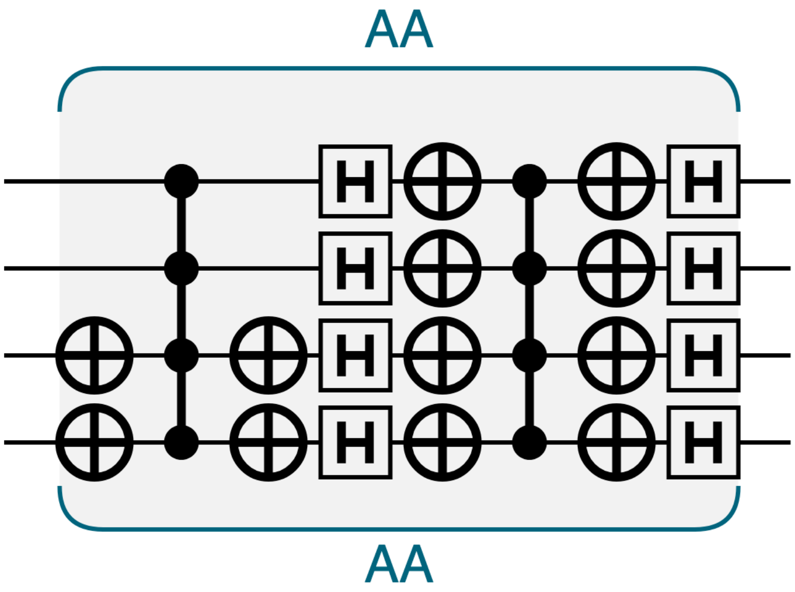
## Intuition on multiple steps

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. This is what we get if we apply the whole flip-mirror combination twice:



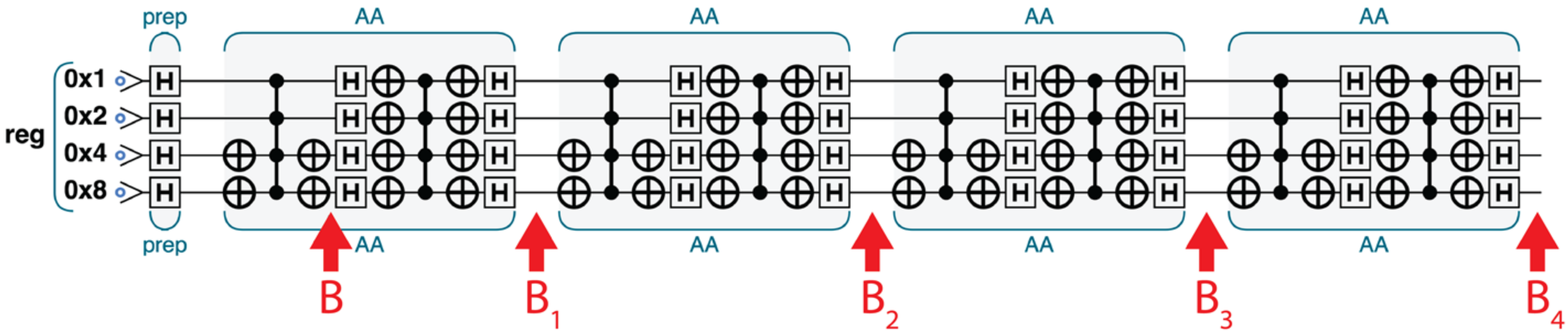
## Single iteration

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. We'll refer to this combined operation as an amplitude amplification (AA) iteration:



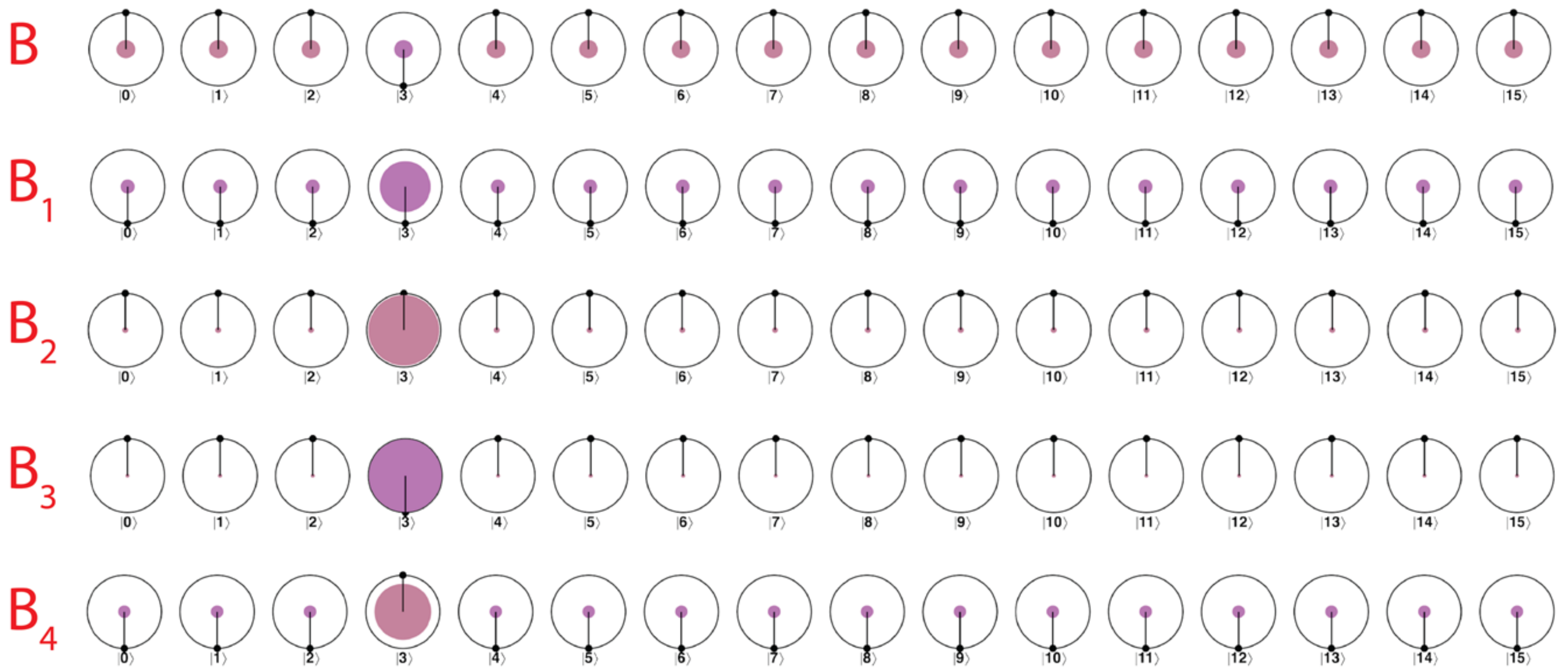
## More iterations

We've applied two AA iterations to the B state, leaving us with a ~90% success probability of observing the marked value. Can we continue applying AA iterations to bring that even closer to 100% ?



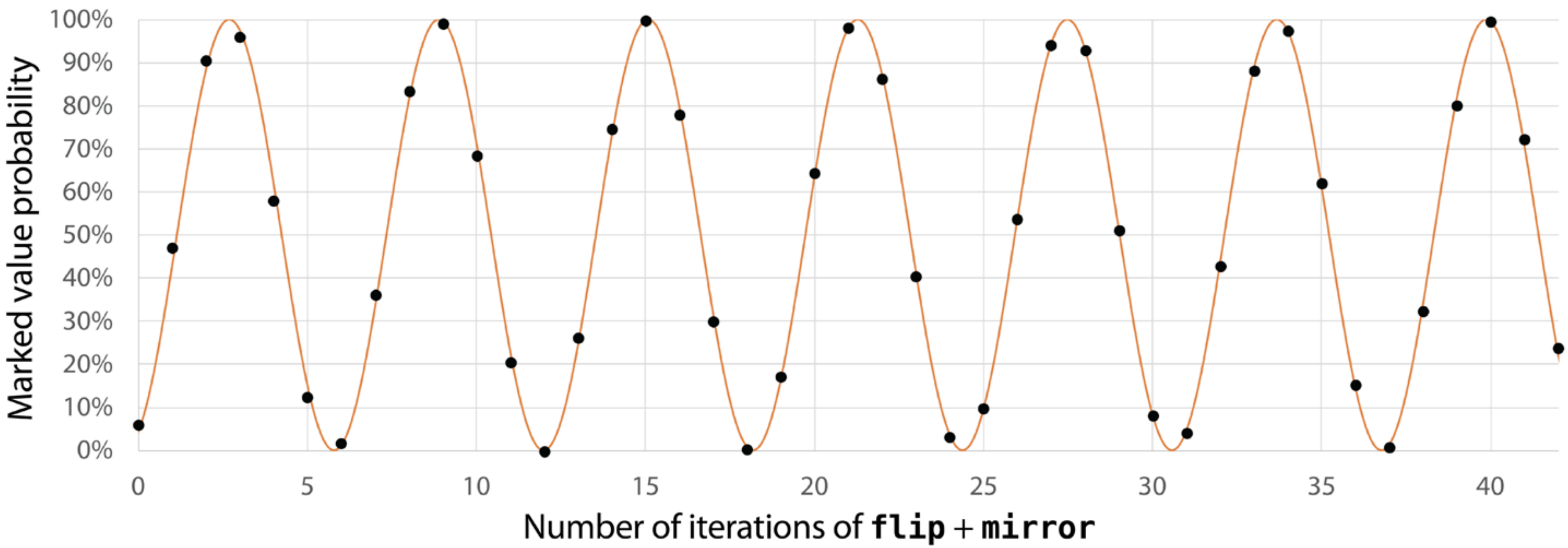
## More iterations

We've applied two AA iterations to the B state, leaving us with a ~90% success probability of observing the marked value. Can we continue applying AA iterations to bring that even closer to 100% ?



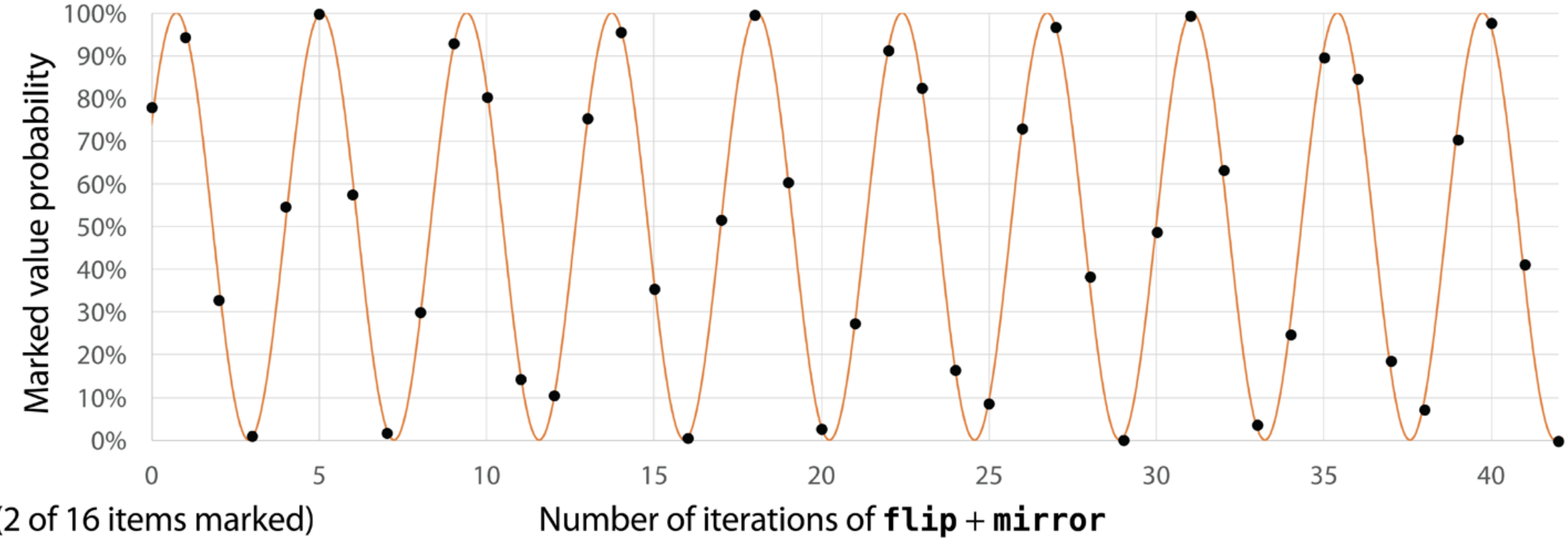
**More iterations**

This plot shows us that as we continually loop through our iterations, the probability of reading the marked value oscillates in a predictable way. We can see that to maximize the chance of getting the correct result, we're best off waiting for the 9th or 15th iteration:



## Two values

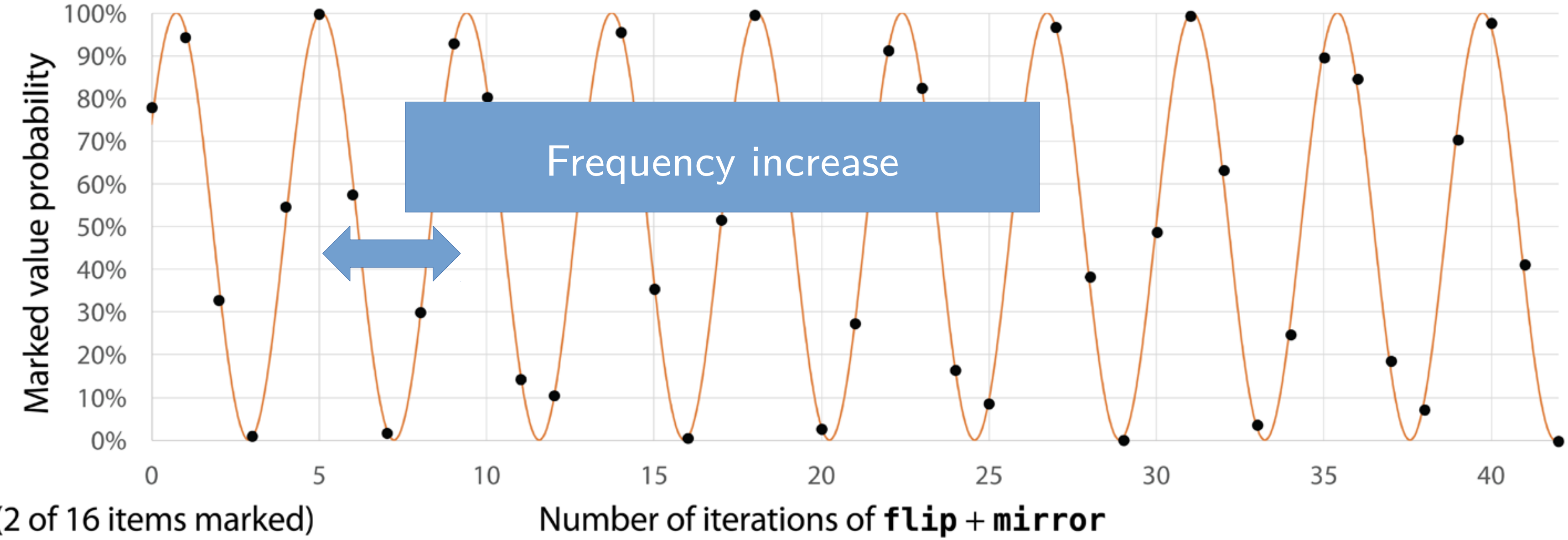
With a small modification to the circuit we can try running multiple AA iterations on a register having any number of phase-flipped values. For instance, let's flip two values:




*Note that the probability shown on the y-axis is the probability of obtaining either one of the (two) marked values if we were to READ our register.*

## Two values

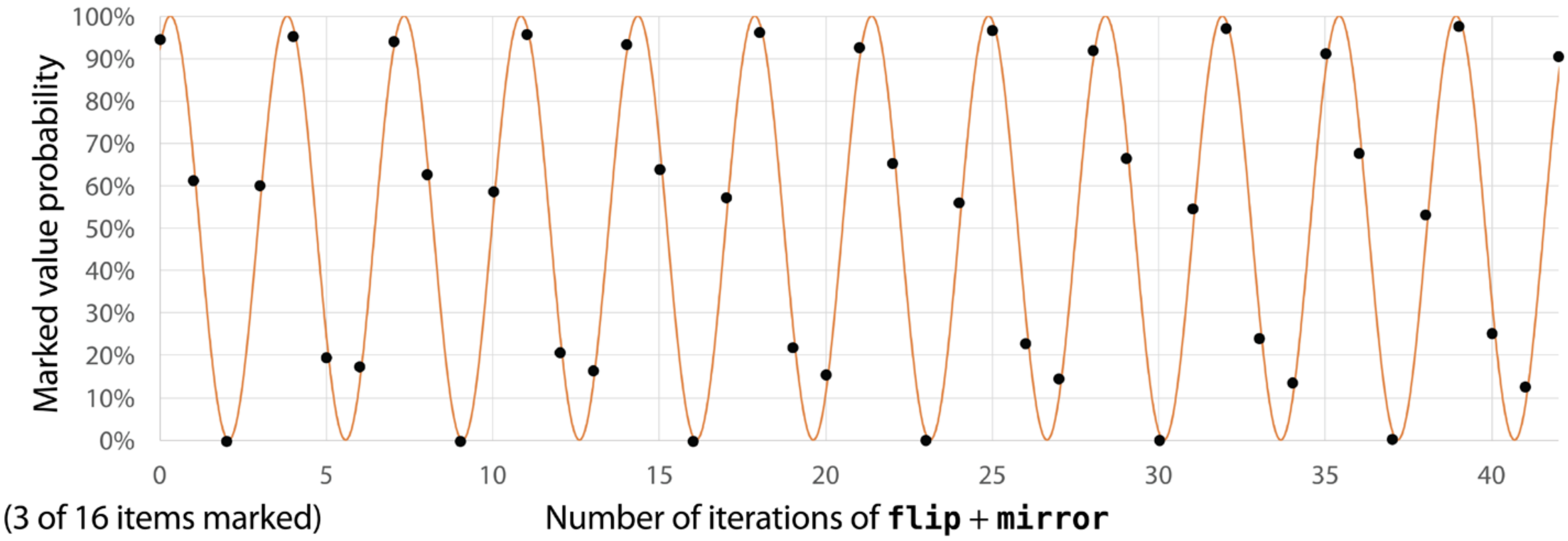
With a small modification to the circuit we can try running multiple AA iterations on a register having any number of phase-flipped values. For instance, let's flip two values:




*Note that the probability shown on the y-axis is the probability of obtaining either one of the (two) marked values if we were to READ our register.*

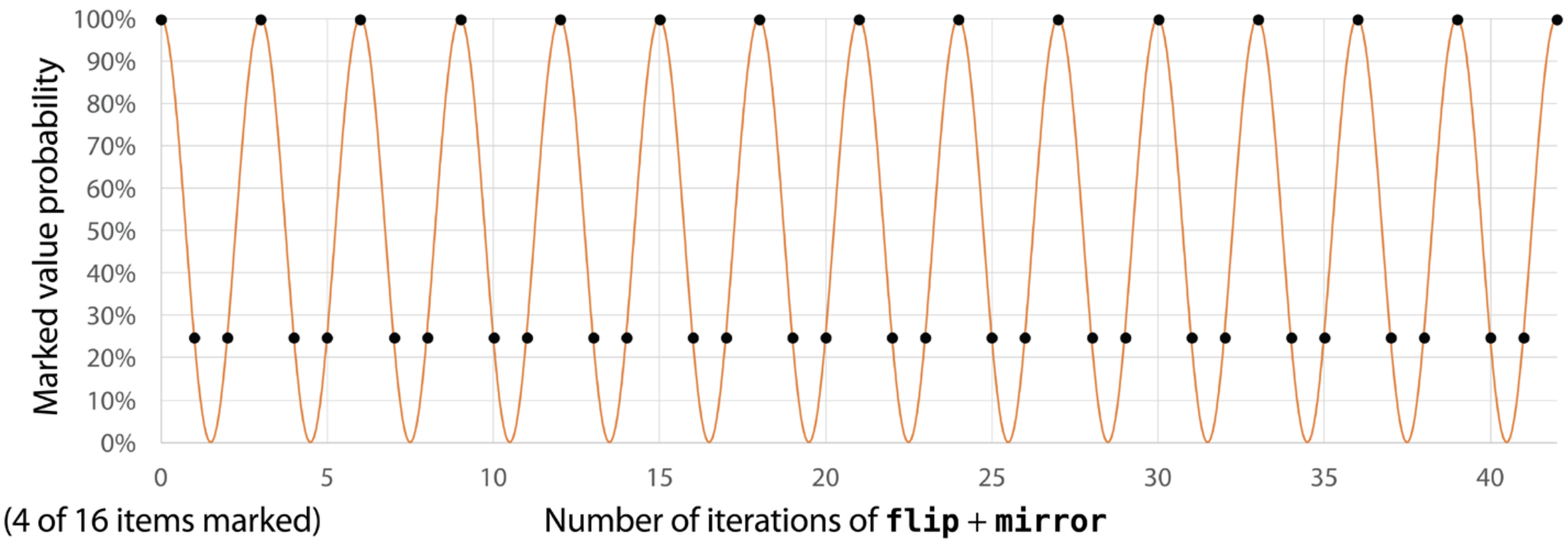
## Three values

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



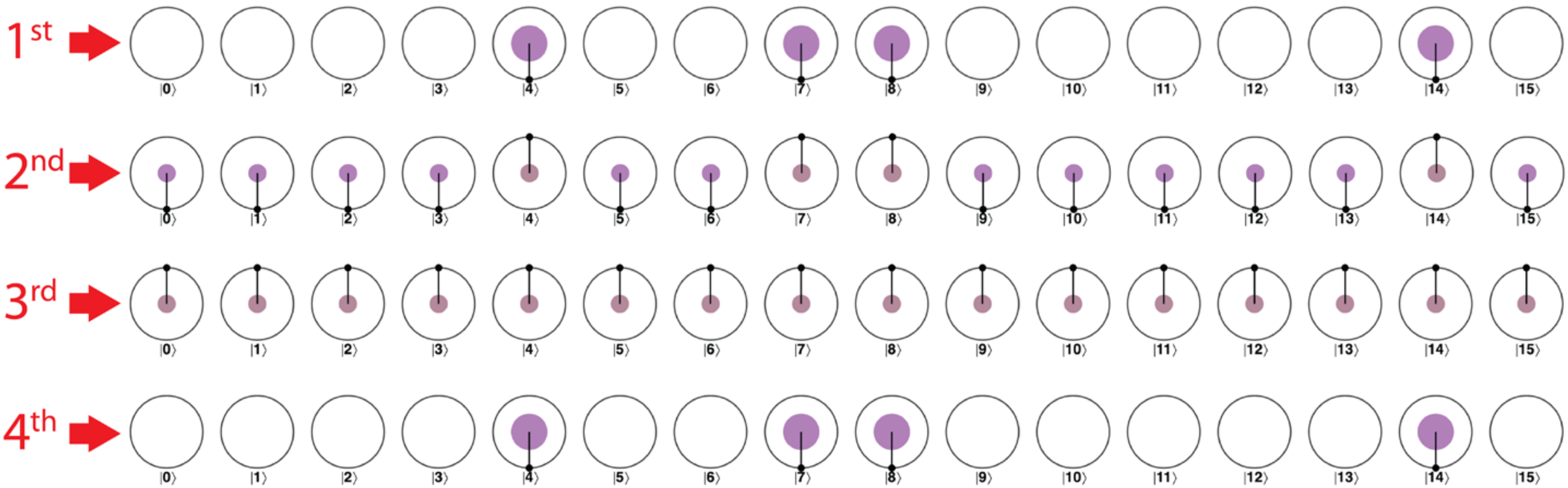
## Four values

When we have 4 of the 16 values flipped something interesting happens. The very first iteration gives us 100% probability of success.



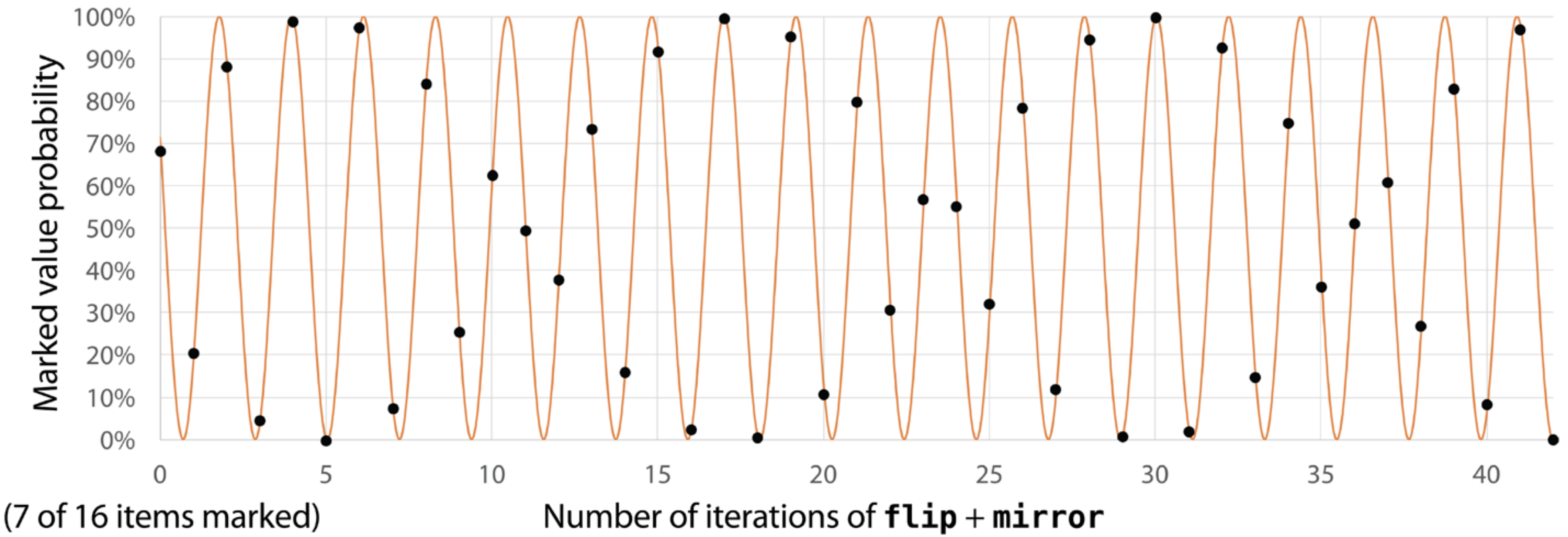
**Four values**

When we have 4 of the 16 values flipped something interesting happens. The very first iteration gives us 100% probability of success.



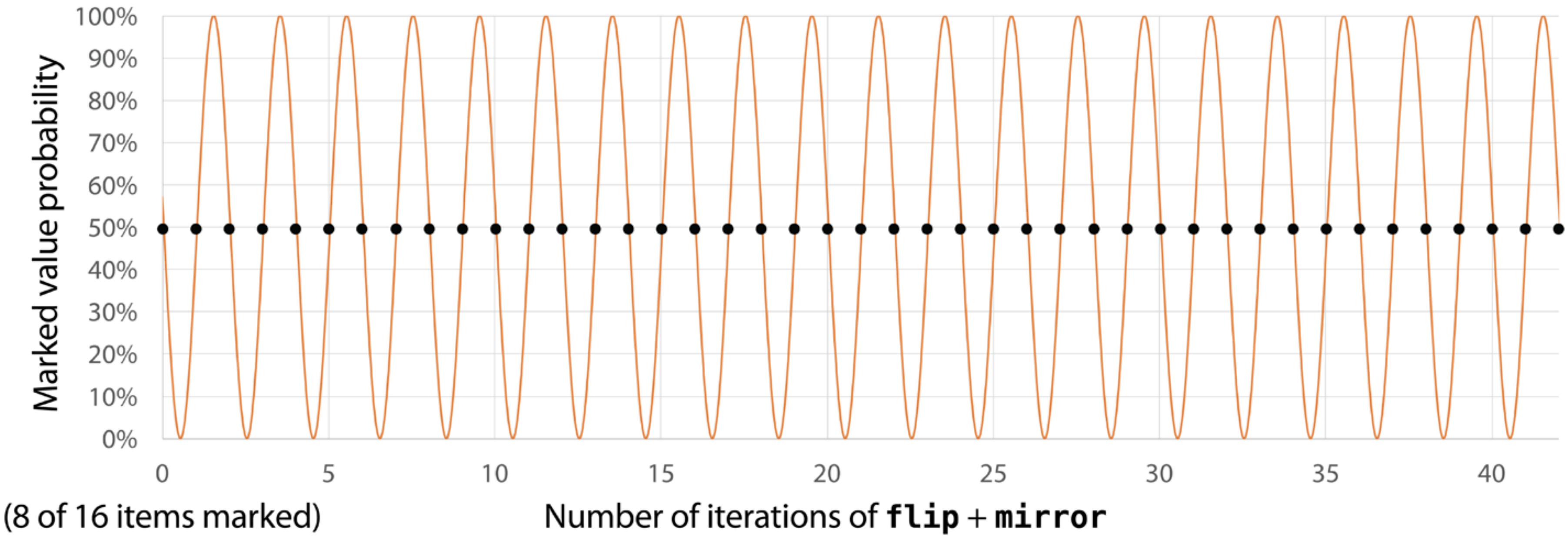
## Seven entries

This trend continues for up to seven flipped values:



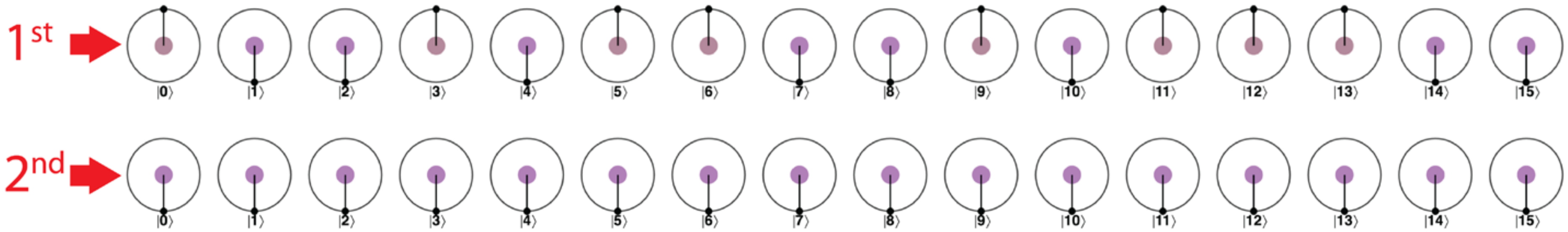
## Eight values

But suddenly stop when eight values are flipped as we are experiencing a global phase (only the relative phase matters for quantum states):



**Eight values**

But suddenly stop when eight values are flipped as we are experiencing a global phase (only the relative phase matters for quantum states):



## Speeding-up conventional algorithms with AA

It turns out that 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*.

## AA and QFT as sum estimation

It turns out that by combining the AA and quantum Fourier transform (QFT) – see next chapter - primitives, we can devise a circuit allowing us to READ not just one of our marked values, but a value corresponding to how many marked values in our initial register state were flipped. This is a form of quantum sum estimation (QSE) – refereed as quantum counting in some books.