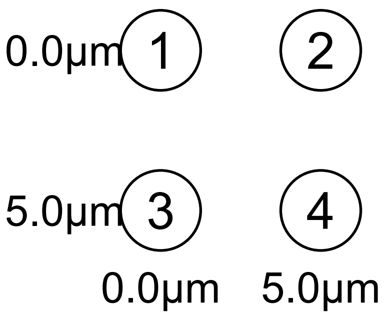
In [1]:

```
# Instaling library
using Bloqade
```

In [2]:

```
# Global variables

C6 = 2*π*862690;
dimension = 2; # 4 atoms at total
r = 5; # distance between 2 neighbors
epsilon = 1.5;
Rb = r + epsilon; # Rydberg blockade distance
Ω = C6/(Rb)^6; # Rabi frequency
t = 3; # simulation time

# Creating the atoms geometry
atoms = generate_sites(SquareLattice(),  dimension, dimension; scale = r)
```

Out[2]:



In [3]:

```
# Declaring the hamiltonian
hamiltonian = rydberg_h(atoms; Ω = Ω)
```

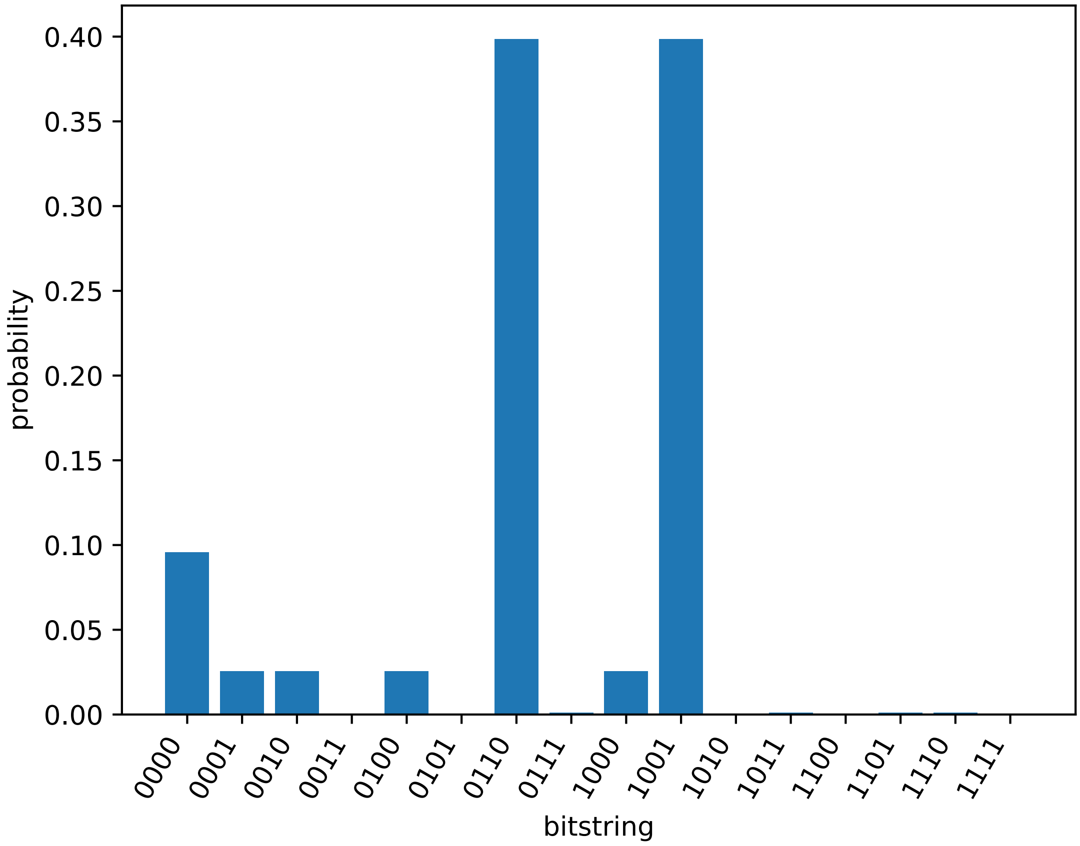
Out[3]:

$$\sum \frac{2\pi\cdot 0.863\times 10^{6.0}}{|x_i-x_j|^6} n_i n_j + 1\cdot 2\pi\cdot 5.72\cdot \sum \sigma_i^x$$

In [4]:

```
# Initial zero vector
reg = zero_state(dimension^2);

# Solving Schrodinger's equation
prob = SchrodingerProblem(reg, t , hamiltonian)
emulate!(prob)

# Finding the most probable reponse
best_bit_strings = most_probable(prob.reg, 1)

# Histogram
bitstring_hist(prob.reg; nlargest = 20)
```

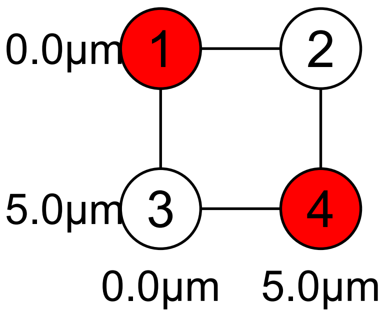
Out[4]:



In [5]:

```
# System final configuration
Bloqade.plot(atoms, blockade_radius = r + epsilon; colors = [iszero(b) ? "white" :
```

Out[5]:

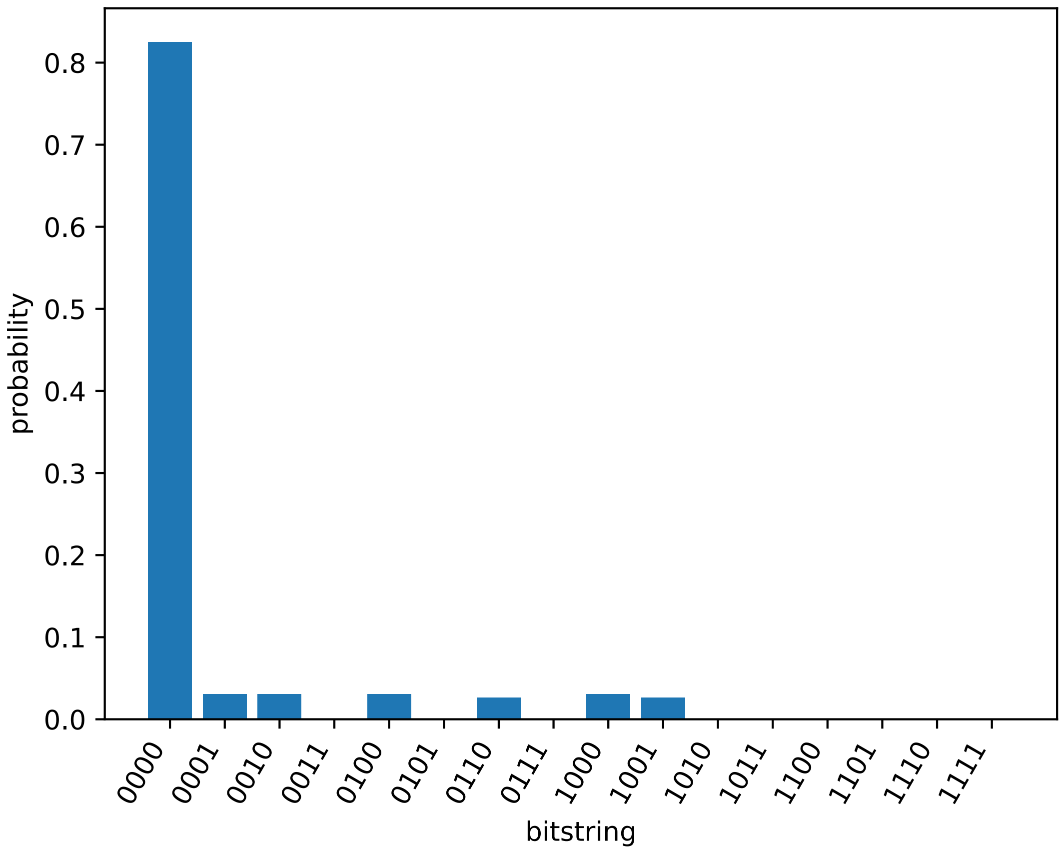


In [6]:

```
# Now we are going to increase the rydberg blockage distance. Note that the diagon
# has a value of sqrt(50) = 7.07
# Let's make the rydberg blocage radius bigger than that

epsilon = 3;
Rb = r + epsilon; # Rydberg blockade distance -- r + epsilon = 5 + 3 = 8 > 7.07
Ω = C6/(Rb)^6; # Rabi frequency

# Declaring the hamiltonian
hamiltonian = rydberg_h(atoms; Ω = Ω);

# Initial zero vector
reg = zero_state(dimension^2);

# Solving Schrodinger's equation
```

```julia
prob = SchrodingerProblem(reg, t , hamiltonian)
emulate!(prob)

# Finding the most probable response
best_bit_strings = most_probable(prob.reg, 1)

# Histogram
bitstring_hist(prob.reg; nlargest = 20)
```

Out[6]:



In [7]:

```julia
# System final configuration
Bloqade.plot(atoms, blockade_radius = r + epsilon; colors = [iszero(b) ? "white" :
```

Out[7]: