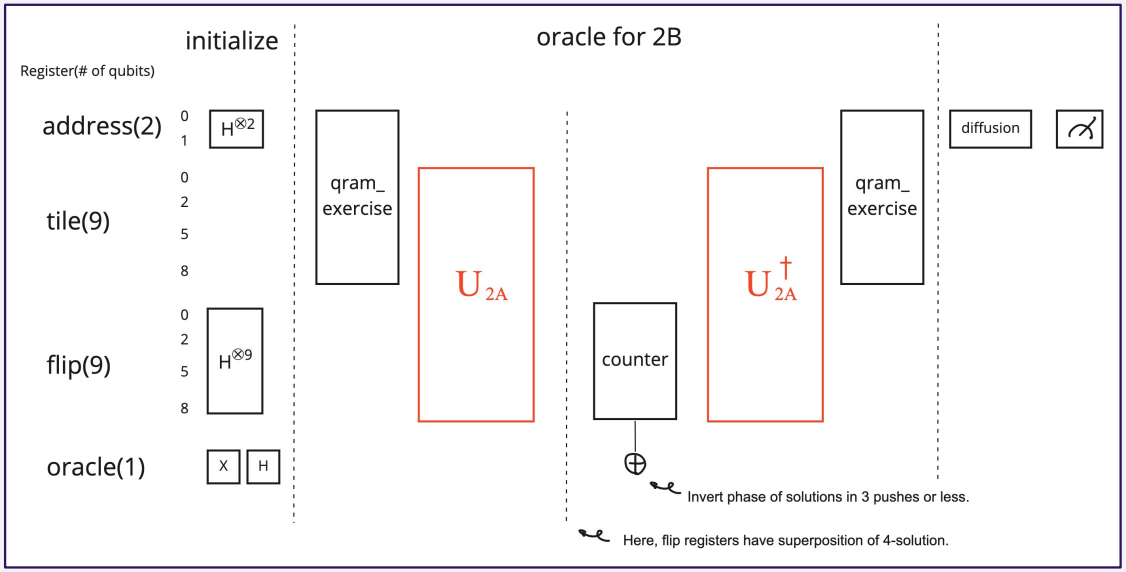
Hint 2-b

Question: I made a unitary circuit U_{2A} that flips the phase according to the board information that is input to tile, but how can we do this when we have 4 boards?



Let‘s refine these implementation examples!
-We can implement U_{2A}† by a reversed circuit of U_{2A}.
-Since Diffusion and Diffusion† (and also QRAM and QRAM†) can be implemented by the same circuit, we omitted their daggers

Dr. Ryoko

The four solutions would be output in a a superposition state, wouldn’t they?
I have taught you how to combine QRAM and the Grover algorithm. You should build a counter that counts up the number of “1”s in your circuit and configure the oracle. Here is a conceptual diagram!

Supplementary material of hint 2-b

Qiskit

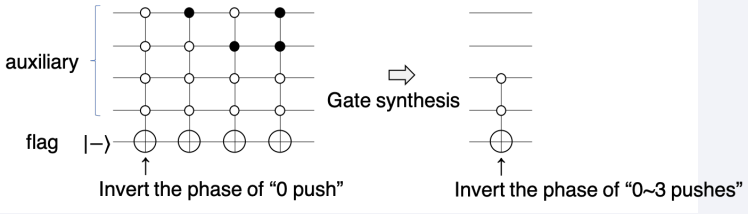
The **counter** in hint 2-b inverts the phase of the "solution" on flip qubits if there are 3 or less "1"s in the solution. *Note: The counter will process the solution of the four boards simultaneously in a superposition state.*

As an example, the counter consists of the following three steps.

1. Outputs the number of "1"s contained in *flip qubits* to the *auxiliary qubit* in binary.

```
def counter(qc, flip,auxiliary):
    for i in range(len(flip)):
        qc.mct([flip[i],auxiliary[0],auxiliary[1],auxiliary[2]],auxiliary[3],mode='noancilla')
        qc.mct([flip[i],auxiliary[0],auxiliary[1]],auxiliary[2],mode='noancilla')
        qc.ccx(flip[i],auxiliary[0],auxiliary[1])
        qc.cx(flip[i],auxiliary[0])
```

2. If the numbers on the auxiliary qubits are 0 to 3, the circuit inverts the phase.



3. Reversed circuit of step 1 (for uncomputation).

Dr. Ryoko

If we adopt Grover's algorithm in U_{2A}, $U_{2A}^{\dagger}$ would look like this...