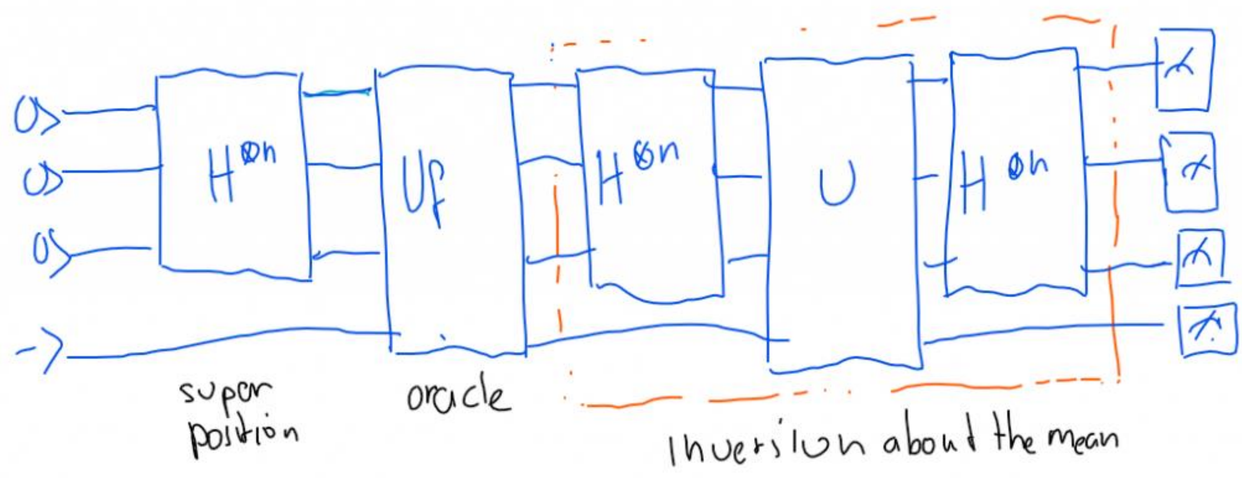
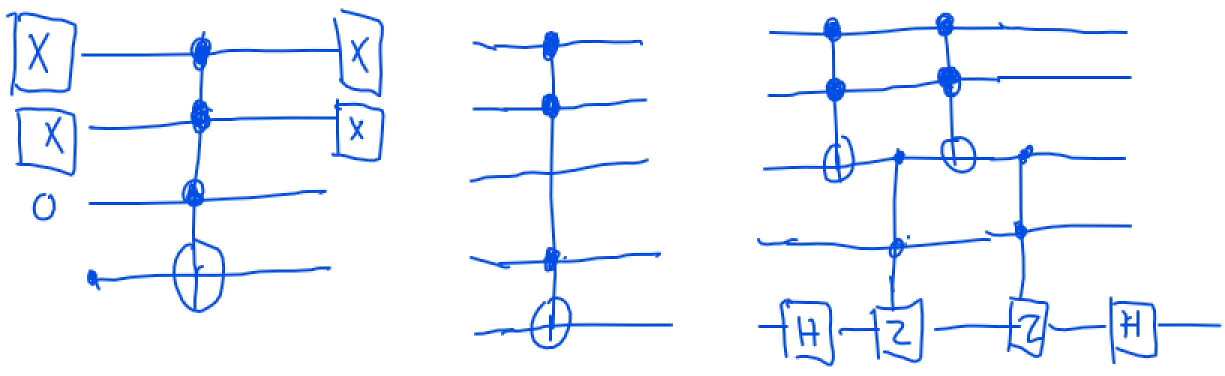
Grover search algorithm with 011_3_qubit_grover_50_qasm

The first step in Grover algorithm is the initialization this bring all the bits (with Hadamard gate) in superposition. Then we have the oracle this defines the search action by means of phase inversion. After the Hadamard's all possibilities are there. The oracle defines the search pattern and will select the right qubit by inverting its phase.

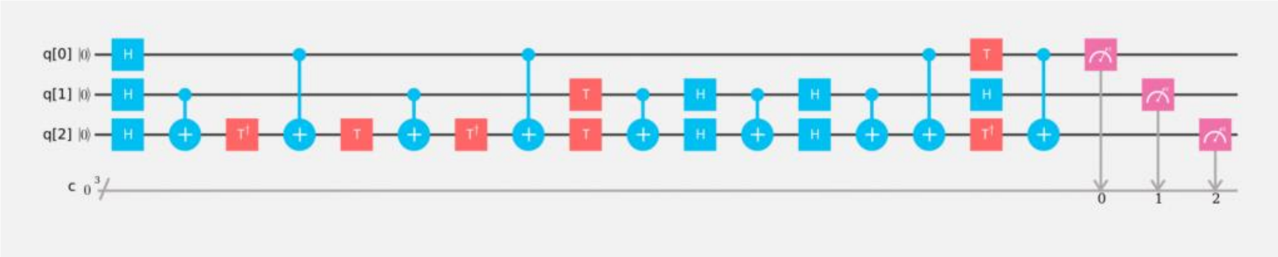
The second step of the Grover algorithm amplifies this result and the answer can be measured. We have an extra ancilla bit starting in the state minus.



As an example, we will start with the search pattern "110". For the oracle Uf and the second U, we will use a Toffoli gate. For the 3-bit input, a 3 qubit Toffoli gate is needed and furthermore, the pattern must be computed and uncomputed.



Now we need 4 qubits, one for the extra ancilla bit, can we do with 3 qubits?



Therefore, we need a Toffoli gate build with basic gates and initialized with Hadamard gate for all qubits. Now the Toffoli gate is in a 3 qubit entangled state.

Deutsch Algorithm

Suppose you have a function f from {0,1} to {0,1}. Decide if f is constant or not. An operator U_f (let's just call it U) operating on two qubits is defined by

```
U|x>|y> = |x>|y XOR f(x)>
```

so it depends on how we choose f. Now with some calculation it can be shown that

```
If f(0) = f(1) = 0 then U|+>|-> = |+>|->
If f(0) = f(1) = 1 then U|+>|-> = -|+>|->
Else if f is not constant then U|+>|-> = |->|->
```

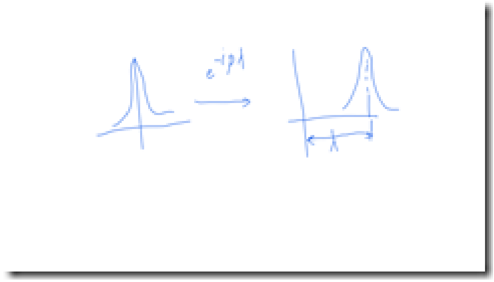
This means that if we set the first qubit to Plus with the Hadamard gate and the second qubit to Minus with Puali X followed by Hadamard, then apply U, we can get to the state PlusMinus (ignoring the global phase) or MinusMinus depending on whether f is constant or not. This with a single call to U. Then we can apply the Hadamard gate again to the first qubit, so we can measure a Zero or One state and produce the output of the algorithm.

You have to encode U for each f. This algorithm looks for value of f(0) xor f(1) to check wether it is balanced or constant. There are two ways. Either you need to have truth table of f, or explicit formula to calculate f(0) and f(1). If you know only the truth table, you can't implement Deutsch algorithm. However, if you know what f(x) looks like, you can build that U gate. Convince yourself that simple CNOT gate is actually equal to the U gate of f(x) = x function.

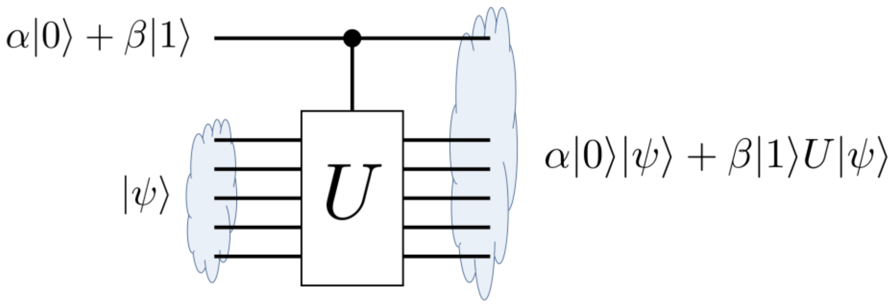
Ipea_3_pi_8_qasm

Phase estimation algorithm or PEA is used for quantum chemistry application, for estimating the energy eigenvalues of a molecular Hamiltonian. The PEA can be used to estimate the value of the phase which allows determining the corresponding eigenvalues of the Hamiltonian.

Phase estimation is a discretization of von Neumann's prescription to measure a Hermitian observable. A measurement is made of a simple observable e.g. the location. We make a convenient observation from quantum mechanics, it's known that a momentum operator p generates shifts for single particles.



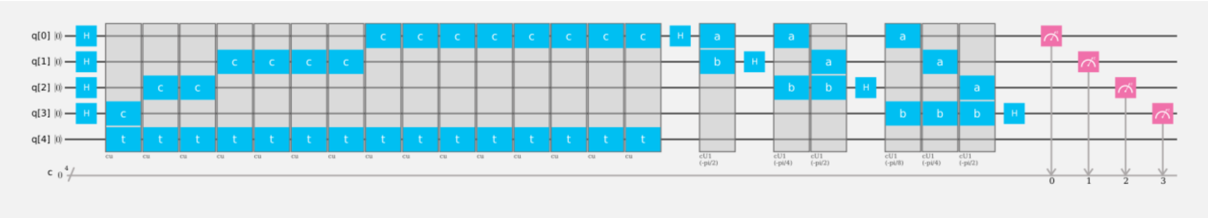
We have a controlled U or Unitary operator. That is, if we apply the unitary to some wave packet, then this wave packet will be shifted by in the positive direction.



In the above circuit the target register is prepared in some state phi. A controlled version of U is a unitary operator acting on the system control and target, where control is a single qubit and target is a register if n qubits. Controlled U applies U to the target register if the control bit is |1>. The control bit gets mapped, while the target register remains in the state phi. Thus we can describe that the action of controlled-U on the composite system control +target by a single qubit phase shift gate P acting on the control bit.

Quantum circuit for phase estimation algorithm

PEA estimates the phase in the eigenvalue of a unitary transformation. A controlled U can be used for a phase kick back circuit.



The (00011) means the estimated phase if read properly. Since your least significant bit (8 CNOTs) is measured in c[0], that is the rightmost digit in your result, so your phase should be 0.0011. The last qubit, q[4], is used to encode the phase on the other four qubits but does not give info about the phase by direct measurement. The total result you obtain is, therefore, 3/16, and the resulting phase should be 2π3/16 or 3π/8.