Hardware simulation of the Deutsch algorithm

Problem statement [edit]

In the Deutsch–Jozsa problem, we are given a black box quantum computer known as an oracle that implements some function:

$$f: \{0,1\}^n \rightarrow \{0,1\}$$

The function takes n-bit binary values as input and produces either a 0 or a 1 as output for each such value. We are promised that the function is either constant (0 on all inputs or 1 on all inputs) or *balanced* (1 for exactly half of the input domain and 0 for the other half).[1] The task then is to determine if f is constant or balanced by using the oracle.

For the remainder, we consider **the case n=1.**

The idea is that the ‘oracle’ gets fed with only one input (either ‘0’ or ‘1’) and decides if the (embedded) function f is or ‘balanced’ (output ‘0’) or ‘constant’ (output ‘1’).

This problem is defined such that a classical computer cannot solve it by using just a single input, as it needs to evaluate **both** ‘0’ **and** ‘1’ to determine if the function f is ‘constant’ or ‘balanced’.

In **[1],** a simulation is described using classic (optical) hardware to mimic (emulate) the Deutsch experiment on a quantum computer. This memo describes the experimental setup of the hardware simulation.

Short Description of the experiment and setup

1. Basis states and superposition

To mimic basis states [0> or [1>, horizontal (0°) or vertical polarization (90°) of a laser pulse is used. Again, it is simulation, not a single quantum particle.

By using a Hadamard gate, a superposition state can be obtained by $1/\sqrt{2}$ ([0> + [1>).
This represents a quantum state, specifically the equal superposition of the basis states $|0\rangle$ and $|1\rangle$. Created by applying a Hadamard gate (H) to the $|0\rangle$ state.
We can do a similar transform on [1>, with result $1/\sqrt{2}$ ([0> - [1>).

Additional notation: a negative expression, like -[0>, represents a horizontal polarization with an 180-degree phase shift, compared to [0>. Similar for –[1>.

1.1 Implementation of superposition in our experiment
A small laser (< 5mW) is used to generate a short (~20 ms) laser pulse.
Choosing input ‘0’ or ‘1’ is done by choosing an orientation of the polarization filter that is installed directly after the laser output.
For input ‘0’, the filter **(I0)** is oriented at 45°. This simulates a superposition state.
For input ‘1’, the filter **(I1)** is oriented at -45°.
Without loss of generality, we continue with input ‘0’, as input ‘1’ is a symmetric case.

2. Function 'f'

Possible functions f0, f1, f2, f3 can be implemented in the oracle.

✅ All Possible $f(x) : \{0,1\} \rightarrow \{0,1\}$

Function	$f(0)$	$f(1)$	Type
f_0	0	0	Constant
f_1	1	1	Constant
f_2	0	1	Balanced
f_3	1	0	Balanced

In our experiment, it would be convenient for implementing 'f' if we could transform a horizontal polarity into a vertical one (and vice versa), independent of input '0' or '1'. Unfortunately, this is not implementable.

To overcome this, a trick is done. For the output value of 'f' to become a '1', a 180° phase shift is applied to the input. For the output value of 'f' to become '0', no phase shift is applied to the input.

2.1 Implementation for the balanced functions

$\mathbf{f_2}(1) = 1$ is implemented by taking the vertical polarization component and apply a 180° phase shift on it, while simultaneously, $f_2(0) = 0$ is implemented by keeping (the phase of the) horizontal polarization component unchanged. For this, we can use a special optical filter: a **half-wave plate** is an optical device that shifts the phase of a wave by half a wavelength (180°).
<u>The net result of $\mathbf{f_2}$ is that the superpositioned state is rotated -90°.</u>

$\mathbf{f_3}(0) = 1$ is implemented by taking the horizontal polarization component and apply a 180° phase shift on it, while simultaneously, $f_3(0) = 0$ is implemented by keeping (the phase of the) vertical polarization component unchanged. For this, we can use the same special optical filter: a **half-wave plate** is an optical device that shifts the phase of a wave by half a wavelength (180°).
<u>The net result of $\mathbf{f_3}$ is that the superpositioned state is rotated +90°.</u>

2.2 Implementation for the constant functions

$\mathbf{f_0}(0) = 0$ and $\mathbf{f_0}(1) = 0$ is implemented by keeping the phase(s) of both polarization components unchanged. This can be implemented by using no filter at all.
<u>The net result of $\mathbf{f_0}$ is that the superpositioned state is rotated 0°.</u>

$\mathbf{f_1}(0) = 1$ and $\mathbf{f_1}(1) = 1$ is implemented by taking both the horizontal and vertical polarization components and apply a 180° phase shift on them. For this, we can use two consecutive **half-wave plates** (90° rotated with respect to each other).
<u>The net result of $\mathbf{f_1}$ is that the superpositioned state is rotated 180°.</u>

3. Measuring

So, the simulated system can determine which function (f0, f1, f2, f3) was used, based on the change in orientation (of the polarization) **and** phase shift (of the polarized laser pulse), of the resulting laser pulse (after the function ‘f’ is applied).

However, in our simple experiment, we can **only measure** the change in **orientation** (of the polarization), not the phase shift (of the polarized laser pulse). In fact, we can detect the two following changes:

Orientation change, 0° / 180°	‘f’ is constant	Apply filter **G0** (*)
Orientation change, -90° / 90°	‘f’ is balanced	Apply filter **G0** (**)

(*)
Filter **G0** is a polarization filter that is applied to the result of function ‘f’, and the filter’s orientation is equal to 45° (in fact, it is <u>chosen</u> to be the same polarization orientation of filter **I0** (see section 1.1)). Since the orientation of polarization is (visibly) <u>not</u> changed, the result is that the filter G0 passes the result of I0 (and ‘f’); and produces a resulting laser pulse (a ‘1’ state)
If we do see a laser pulse at the end, the function is constant

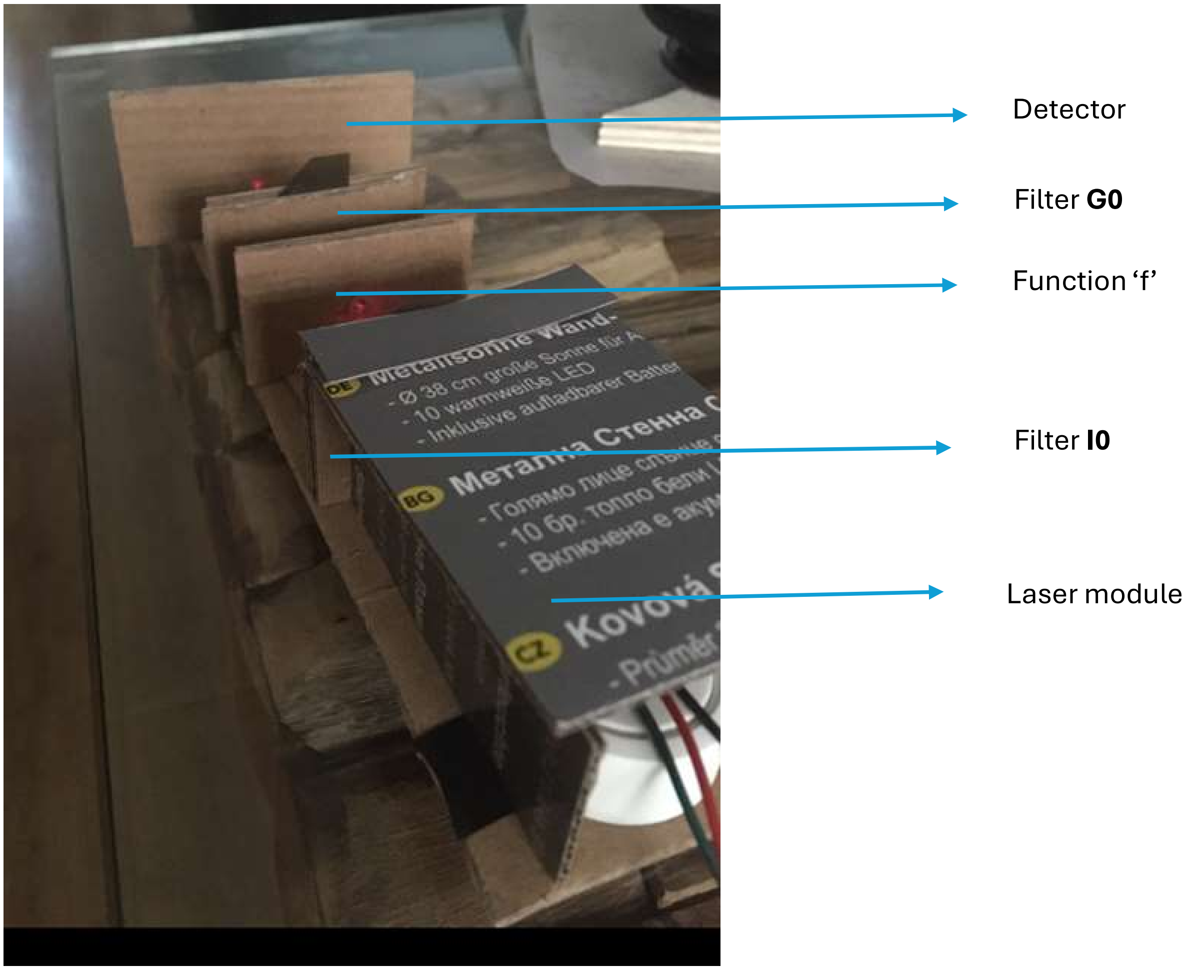
(**), similar as for (*):
Since the orientation of polarization is <u>changed</u> by +/- 90°, the result is that the filter G0, as define above, blocks the resulting laser pulse; and produces <u>no</u> laser pulse (‘0’ state)
If we do <u>not</u> see a laser pulse at the end, the function is balanced

Note:

Alternatively, after applying function ‘f’, in theory, the polarization could be rotated 45°, such that the laser pulse has either a purely horizontal or vertical polarization (in the frame of reference). Then the measurement (e.g. apply a horizontal polarization filter instead of G0) “collapses” the polarization of the laser pulse to a basis state.

4. Hardware implementation

Experimental setup (breadboard) with laser compartment and filters I0 (directly after the laser), function ‘f’, and filter G0, as well as a detector (card at the end).



5. References

[0] Youtube movie: https://www.youtube.com/watch?v=muoIG732fQA

[1] Youtube movie: https://www.youtube.com/watch?v=tHfGucHtLqo