Physics Letters A 380 (2016) 1925–1931

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Quantum computation with classical light: Implementation of the Deutsch–Jozsa algorithm

Benjamin Perez-Garcia [a,b,c], Melanie McLaren [b], Sandeep K. Goyal [c,e], Raul I. Hernandez-Aranda [a], Andrew Forbes [b], Thomas Konrad [c,d,*]

[a] Photonics and Mathematical Optics Group, Tecnológico de Monterrey, Monterrey 64849, Mexico
[b] University of the Witwatersrand, Private Bag 3, Johannesburg 2050, South Africa
[c] School of Chemistry and Physics, University of KwaZulu-Natal, Private Bag X54001, Durban 4000, South Africa
[d] National Institute of Theoretical Physics, Durban Node, Private Bag X54001, Durban 4000, South Africa
[e] Institute of Quantum Science and Technology, University of Calgary, Alberta T2N 1N4, Canada

ARTICLE INFO

Article history:
Received 6 October 2015
Received in revised form 4 April 2016
Accepted 4 April 2016
Available online 8 April 2016
Communicated by P.R. Holland

Keywords:
Deutsch–Jozsa algorithm
Quantum computation
Classical light
Binary tree

ABSTRACT

We propose an optical implementation of the Deutsch–Jozsa Algorithm using classical light in a binary decision-tree scheme. Our approach uses a ring cavity and linear optical devices in order to efficiently query the oracle functional values. In addition, we take advantage of the intrinsic Fourier transforming properties of a lens to read out whether the function given by the oracle is balanced or constant.

1. Introduction

Quantum computation has reached a stage wherein concepts and theory are properly understood while its implementation has not yet surpassed the proof-of-principle level. The difficulties lie in the state preparation and coherent control of a multitude of two-level systems carrying basic units of quantum information known as *qubits*. In particular, decoherence due to the leak of information to the environment is a problem. Even so, quantum computation promises unprecedented computational power and the possibility to tackle hitherto unsolvable computational tasks. This is done by simultaneously processing a multitude of numbers encoded in large superpositions of the corresponding states of quantum systems. There are several quantum algorithms that have been experimentally realised. For instance, the Deutsch [1] and Deutsch–Jozsa [2] algorithms have been implemented in nuclear-magnetic-resonance systems [3–8], QED cavities [9–12], quantum dots [13,14], trapped ions [15], light shifts [16], superconducting quantum processors [17], nitrogen-vacancy defect centre [18] and quantum optical systems [19–22]. However, due to the aforementioned problems, the maximum number of qubits used in such implementations has not yet exceeded a few. For example, the greatest number of qubits used for the Deutsch–Jozsa Algorithm was four [23]. Thus there has not yet been a computational problem solved on a quantum computer which was inaccessible for classical computers (Turing machines).

Here we propose a candidate for the first scalable implementation of the Deutsch–Jozsa Algorithm. A salient feature of our proposal is the use of the spatial degree of freedom of classical light fields, allowing to efficiently encode a restricted class of functions. After the Deutsch Algorithm [24], this is the second scheme we present of quantum computation with classical light and the first one that might outperform any Turing machine.

2. Deutsch–Jozsa algorithm

The Deutsch–Jozsa Algorithm aims to distinguish between binary functions $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$ that are constant, $f(x) = 0$ or $f(x) = 1$ for all arguments x, and balanced functions f which assume for $N/2$ of their arguments the value 0 and for the other $N/2$ of arguments the value 1. The problem cannot be solved efficiently on Turing machines if the number N is very large, the reason lies in the need to inquire up to $N/2 + 1$ functional values from an oracle (or databank). For example, let the arguments of the function be stored in an input register of 200 Bits and run

[*] Corresponding author. Correspondence to: School of Chemistry and Physics, University of KwaZulu-Natal, Private Bag X54001, Durban 4000, South Africa.
E-mail address: konradt@ukzn.ac.za (T. Konrad).

http://dx.doi.org/10.1016/j.physleta.2016.04.006
0375-9601/© 2016 Elsevier B.V. All rights reserved.

over the full possible range from 0 to $2^{200} \approx 10^{60}$. If every query for a functional value would just take a picosecond then evaluating $N/2+1$ values would require up to 10^{47} s, a large multiple of the age of the universe. A quantum computer can access and process all the functional values at the same time based on the superposition principle and yield the right result using interference between the processed states representing the functional values.

We note that it has been argued [25] that the Deutsch–Jozsa problem can be solved on a Turing machine with a finite success probability in polynomial time by checking only a small fraction of functional values. In contrast, the present implementation scheme aims to find with certainty the correct answer. It uses a version of the Deutsch–Jozsa Algorithm [26] which works only with a single register with $n = \log_2 N$ qubits as explained in the following. Initially each of n qubits in the input register prepared in state $|0\rangle$, is processed by a Hadamard transformation, $|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, yielding a superposition of states, which corresponds to 2^n arguments x:

$$|\Psi_{\text{in}}\rangle = |0, 0, \ldots, 0\rangle$$
$$\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x_1,\ldots,x_n=0}^{1} |x_1, \ldots x_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \, . \qquad (1)$$

Here $x_1, \ldots x_n$ are the binary digits representing the number x. Thereafter the value 0 (or 1) of $f(x)$ is encoded as relative phase factors $+1$ (or -1) of the input state $|x\rangle$:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \, . \qquad (2)$$

Because the transformation $|x\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle$ is linear it imprints the relative phases carrying the information about $f(x)$ for all x-values in the superposition $\sum |x\rangle$ at the same time. A second Hadamard transformation of the n qubits

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \xrightarrow{H^{\otimes n}} \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{z=0}^{2^n-1} (-1)^{x \cdot z + f(x)} |z\rangle \, , \qquad (3)$$

yields the result of the computation as an interference effect. The term $x \cdot z$ denotes the bitwise inner product of x and z. If the function is constant, the second Hadamard transformation reverses the first one and propagates the system back into its initial state $|0, 0, \ldots, 0\rangle$. This can be interpreted as constructive interference of the equal phase factors forming the probability amplitude of state $|0, 0, \ldots, 0\rangle$, cp. right-hand side of (3). On the other hand, a balanced function results in the same number of positive $(+1)$ and negative phase factors (-1) which destructively interfere to a zero probability amplitude for the state $|0, 0, \ldots, 0\rangle$. Whether the function is constant or balanced can thus be determined by measuring if $|0, 0, \ldots, 0\rangle$ is populated or a state in its orthogonal complement.

The central transformation (2), which encodes all functional values as binary phases for each argument x into the register, has an immediate optical analog – the imprint of position-dependent phases on the electric field on a transversal plane using a phase mask. A corresponding optical implementation of the Deutsch–Jozsa algorithm [27] realised the phase imprint by means of a spatial light modulator (SLM) which acts at a transversal plane on plane-wave laser light. The subsequent Hadamard transformation (3) is realised by means of the Fourier-transform properties of a thin lens (see Appendix A). However, such a straightforward encoding is based on the unary representation of each number x corresponding to a position on the transversal plane, i.e., the phase shifts are encoded on the SLM argument by argument. This procedure requires therefore as many steps to encode functional values in form of phase shifts as there are arguments x. Hence the function cannot be written efficiently into the register and the solution fails, if the number of arguments is large.

3. Binary tree design

There are several ways to model quantum computation, for instance, the quantum circuit model and measurement-based quantum computation [28,29]. Although binary tree designs have been previously reported [30,31], here we propose a novel binary decision tree design as a new paradigm to model quantum algorithms which can be helpful to provide an efficient classical optics picture of such algorithms (see Figs. 1–3). The method we present here combines the efficient creation of the superposition of all arguments $x \in \{0, 1, \ldots, 2^n - 1\}$ with the encoding of the functional values in the form of phase factors. It is based on the optical approach of Daniela Dragoman [32] to prepare a superposition of 2^n states $|x\rangle$ efficiently, i.e., in n steps using a binary decomposition of the number x. Our approach modifies Dragoman's idea in order to implement the Deutsch–Jozsa algorithm by means of path qubits and a ring resonator.

The principle of the method is sketched in Fig. 1. In each step of the preparation corresponding to a round trip in the ring cavity, one additional path qubit is generated by splitting each of the present light beams, i.e., doubling the number of beams. In the

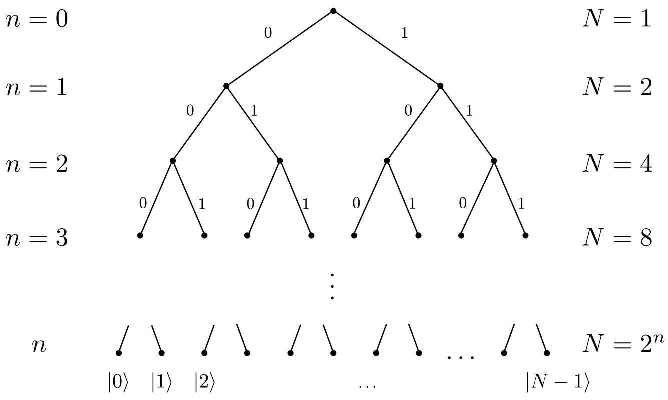

Fig. 1. Diagrammatic representation of scheme to prepare a superposition of 2^n states in n steps. It resembles the quantum walk on a graph with two edges at each vertex (a binary tree).

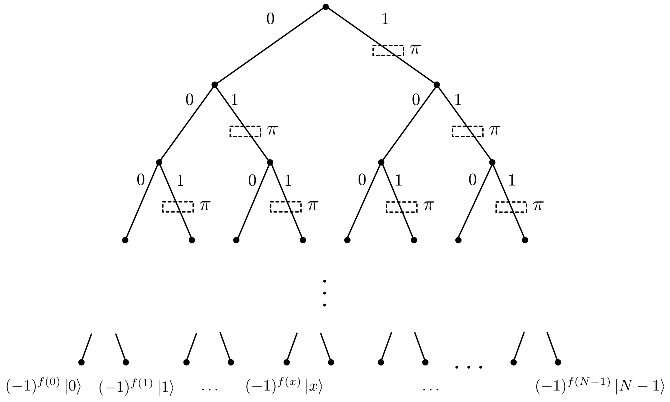

Fig. 2. Any binary function f with $N = 2^n$ arguments x can be encoded in the superposition $\sum(-1)^{f(x)}|x\rangle$ by means of individual π phase shifts in the branches labelled "1". Moreover, all phase shifts on a given level of the tree can be implemented simultaneously by a single operation in the realisation scheme.

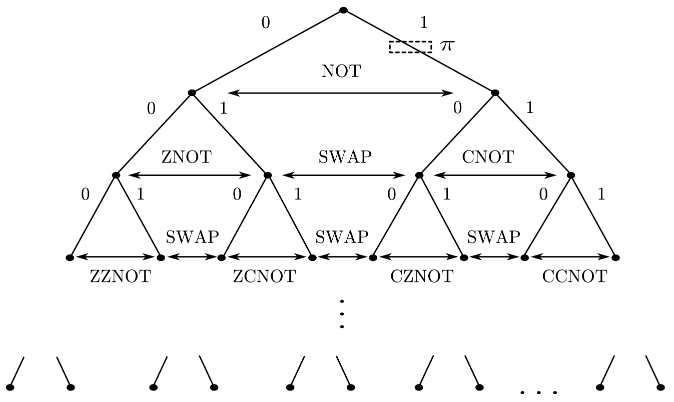

Fig. 3. Swapping states $|0\rangle$ and $|1\rangle$ on different levels of the binary tree corresponds to Not and controlled Not operations at different depth. For example, a swap on the third level given by $|110\rangle \leftrightarrow |111\rangle$ corresponds to a CCNOT operation, also know as Toffoli gate. Each balanced function can be encoded by a specific combination of these swaps and a π phase shift.

first step, a single path qubit is created

$$|\psi_{\rm in}\rangle = |0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\,, \tag{4}$$

where $|0\rangle$ ($|1\rangle$) represents the left (right) part of the beam (cp. Fig. 1). In the second step, each of the beams is split again:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \to \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)\,. \tag{5}$$

We denote by $|x_1, x_2\rangle \equiv |x_1\rangle\,|x_2\rangle$ the part of the light beam that propagates in direction $x_1 \in \{0,1\}$ and $x_2 \in \{0,1\}$ after the first and second step, respectively. In n steps the state thus changes according to

$$|\psi_{\rm in}\rangle \to \frac{1}{\sqrt{2^n}} \sum_{x_1,\ldots,x_n=0,1} |x_1,\ldots x_n\rangle\,. \tag{6}$$

This procedure generates 2^n localised light fields, represented by the state $|0\rangle + |1\rangle + \ldots + |2^n - 1\rangle$, in only n steps, i.e., efficiently. Moreover, our ring cavity setup achieves this with a constant small number of optical devices. Transformation (6) thus yields the input state for the central transformation (2) of the Deutsch–Jozsa algorithm above. Instead of encoding the information about the function f separately, cp. transformation (2), we combine it with the efficient preparation of n qubits (6):

$$|\psi_{\rm in}\rangle \to \sum_{x=0}^{2^n-1} (-1)^{f(x)}\,|x\rangle\;. \tag{7}$$

A subsequent Fourier transform by means of a thin lens interferes the electric fields in the focal point (see Appendix A) resulting in zero intensity for balanced functions and a finite intensity for unbalanced ones, including constant functions. The intensity is a measure of how biased the function is, but this is not relevant for the Deutsch–Jozsa problem.

It is in principle possible to realise (7) for any of the 2^N binary functions f with $N = 2^n$ arguments by applying only phase shifts to beams propagating to the right, cp. Fig. 2. Alternatively, (7) can be imposed using phase shifts combined with controlled bit flips, cp. Fig. 3.

However, to encode an arbitrary function, the path qubits have to be addressable individually, i.e., the light spots on each level of the tree diagram must not overlap. For paraxial monochromatic light, which has finite beam waist, this means that the spatial spread of the tree grows proportional to the number N of arguments. The size of such a computer would thus scale exponentially with the number n of qubits used. For example for violet light of wavelength $\lambda = 400$ nm, to avoid overlap our implementation scheme would thus be restricted to about $n = 23$ round trips, as the diffraction limit imposes a minimum spot size of the order of magnitude of the wavelength. The approximated spot size $\delta \approx 4 \times 10^{-7}$ m is inserted into Eq. (B.3) to obtain the maximum number of qubits $n = 23$, and therefore $N \approx 8 \times 10^6$ states.

On the other hand, certain functions can be programmed without separate access to every beam and hence tolerate overlap. In the following we want to restrict our implementation to such cases. It is remarkable that in our scheme the configuration of light spots would allow in principle to read out the full information about a function even in the case where many light spots overlap – as long as no two spots overlap completely (Appendix C). Moreover, the realisation scheme given here distinguishes constant from balanced functions with partial overlap with the same accuracy as without overlap, see Appendix D.

Normally, encoding the information about f as in transformation (2) is outsourced to an oracle, i.e. a black box. Here we take the standpoint that it must be possible to encode f efficiently, otherwise the algorithm does not yield a result for a large number N. To encode constant functions in our scheme is trivial. In case $f(x) = 0$ for all arguments x, the transformation (7) is identical to transformation (6). A global phase factor of (-1) appears in case $f(x) = 1$ for all x, but can be omitted because it is physically irrelevant.

On the other hand, to generate transformation (7) for the $\binom{N}{N/2}$ balanced functions is in general non-trivial. However, our scheme allows to implement balanced functions which correspond to product states efficiently by choosing whether to apply to each newly generated qubit a relative phase shift:

$$\sum_{x_1,\ldots,x_n=0,1} (-1)^{f(x_1,\ldots,x_n)}\,|x_1,\ldots,x_n\rangle$$
$$= \pm(|0\rangle \pm |1\rangle)(|0\rangle \pm |1\rangle)\cdots(|0\rangle \pm |1\rangle)\,. \tag{8}$$

This class of functions comprises 2^{n+1} elements corresponding to all possible combinations of relative phase shifts in n qubits as well as a global phase shift. As argued in the discussion (Sec. 5), in principle, i.e., given suitable sources and optical devices for light of sufficiently small wavelengths, our implementation scheme allows to encode high numbers n of qubits in n steps and therewith $N = 2^n$ values of constant or balanced functions of the type (8). Moreover, it can distinguish constant from balanced functions in a single step. This is not possible with any classical Turing machine and was regarded before as only attainable in the quantum realm.

A small fraction of all balanced functions f corresponding to entangled states $\sum(-1)^{f(x_1,\ldots,x_n)}\,|x_1,\ldots,x_n\rangle$ can be programmed efficiently in our setup by the help of phase masks and controlled bit flips. Moreover, a relabelling scheme yields all balanced functions carrying out the algorithm without changing the physical realisation. The relabelling procedure can depart from any state with equal distribution of $+1$ and -1 phase factors over the localised light fields and implements a desired balanced function by suitably changing the labels of the light fields. As the renaming does not have any physical implication, the subsequent Fourier transform of the electric field yields zero intensity on the optic axis in the focal plain indicating a balanced function. Therefore, the renaming can be omitted, and any balanced function of 2^n arguments can be programmed efficiently by applying a single relative π phase shift on the first of n path qubits.

A major advantage of the scheme using classical states of light is that it can solve a generalisation of the Deutsch–Jozsa problem efficiently, which has no analogous solution on a quantum computer, let alone on a Turing machine. The "Generalised Deutsch–Jozsa algorithm" introduced here, aims to distinguish not only between constant and balanced functions, but also between functions with different imbalance. Our scheme can detect the number of functional values equal to one encoded by the oracle by measuring the intensity. The programming of a restricted class of functions with an arbitrary number M of ones between 0 (no ones) and $N = 2^n$ (only ones) by the oracle can be realised efficiently (in n steps) by suitably modifying the scheme used for the Deutsch–Jozsa algorithm. The distinction of these functions is carried out, given light of sufficient intensity, in a single step by a Fourier transforming lens as in the Deutsch–Jozsa algorithm. The generalised algorithm and its realisation is discussed in Appendix E. In passing we note, that in order to accomplish the same distinction by means of quantum systems, $O(N)$ repetitions would be required, where N is the number of functional values. The possibility to distinguish an arbitrary number of functions efficiently shows that we can encode in a classical light field an unlimited amount of information and read it out efficiently.

4. Realisation scheme

Our implementation scheme to generate the superposition of 2^n states (6) efficiently, based on Dragoman's approach [32], is shown in Fig. 4. A particular feature of our implementation is that it is based on classical light fields. One should note that the characteristic ingredients of quantum computation are present not only in quantum mechanics but also in classical optics. In particular, superposition, interference and a classical analog of entanglement [33–37] can be produced using coherent laser sources and linear optical elements. Classical optics has been employed to simulate quantum gates [32], to realise quantum walks [38,39] as well as to implement the Deutsch Algorithm [24] and the Deutsch–Jozsa Algorithm [27] with classical states of light. On the other hand, classical light is used in optical computing, e.g., to parallelise vector-matrix multiplication in order to solve certain instances of the travelling-salesman problem [40].

In the implementation, a qubit is created by a pair of slits that are illuminated by a pulsed laser source. These slits can be opened, closed or covered partly by a phase plate. For instance, to prepare a superposition of the form $(|0\rangle + |1\rangle)/\sqrt{2}$ both slits are opened. The state $(|0\rangle - |1\rangle)/\sqrt{2}$ can be prepared by covering one slit of a pair with a π phase plate. In order to understand the principle of the implementation scheme, let us consider the first round trip inside the ring cavity. We assume that all slits are open. Due to the combination of the slits S1 and S2, four distinct light dots are created as indicated in Fig. 5(b). The transversal light field and thereby the four dots are rotated by a Dove Prism (DP) as shown in Fig. 5(c). Then the light field is passed through a cylindrical lens system L1 and L2. The combination of these lenses produces four light stripes as depicted in Fig. 5(d). Finally, another pair of slits (S3) generates 8 light dots, as sketched in Fig. 5(e), by intersecting the four stripes. An additional $4f$-system of spherical lenses L3 and L4 images the transversal plane at the slits S3 to the transversal plane at the entrance of the Dove prism. These steps can be repeated in order to create more dots. Note, that in each round trip the number of dots doubles. Furthermore, in order to minimise the overlap between dots, the distance between adjacent dots can be halved by changing the orientation of the DP. In addition, this procedure keeps the illuminated area of the light field on a transversal plane after each round trip constant (see Appendix B).

The binary function f is programmed in the presented setup as follows. For the constant functions $f(x) = 0$ the setup shown in Fig. 4 is used without any additions. For the alternative constant function, $f(x) = 1$ for all arguments x, the same setup can be used, since a global phase factor of -1 of the electric field, does not make any detectable difference. For the balanced functions the setup is augmented by a π-phase in front of a one slit of a pair of slits. The slits can be addressed because of their separation individually. By adding more phase plates in front of other slits or in particular round trips, it is possible to construct all balanced functions that belong to product states (8) to which we restrict our study in the following. Finally, the output of the cavity is Fourier transformed [41] by a lens (L5) and captured in a CCD camera. Light detected in the focal point behind the lens implies that the programmed function is constant, otherwise, if no light is measured the function is balanced (see Appendix A).

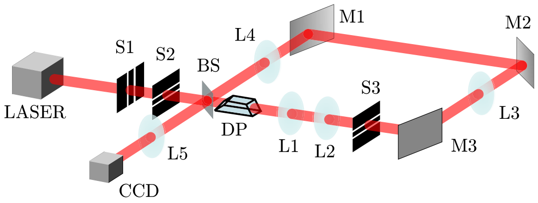

Fig. 4. Experimental implementation. S1–S3: slits; BS: beam splitter; DP: Dove prism; L1 and L2: cylindrical lenses; L3 and L4: $4f$ imaging system, L5: Fourier transforming lens; M1–M3: mirrors.

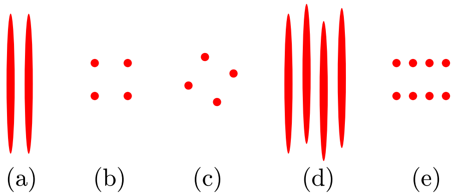

Fig. 5. Output after: (a) S1, (b) S2, (c) DP, (d) L2, (e) S3.

The present implementation scheme determines whether a programmed function is balanced or constant, by integrating the electric field, which carries the functional values as phase factors ±1, over a transversal plane. This integration is accomplished by a thin circular lens which superimposes the light from the focal plane in front of the lens in the focal point behind the lens. The result of the integration stays the same, even if part of the light is superposed already during its passage through the ring cavity before the lens, as explained in Appendix D. Therefore, the scheme tolerates partial overlap between light spots when they are generated.

5. Discussion

The main problems of optical simulations of quantum computation are pointed out in Ref. [42]: the exponential growth of the number of optical devices with the number of qubits, the exponential growth in space (cross section of the light field) and the decrease of optical power. The present implementation scheme solves the first problem by the repeated use of the same optical elements in a ring cavity. The increase of the cross section of the light field, which results from doubling the number of light dots, can be constrained either by (i) adopting the orientation of the DP after each round trip (Appendix B) or by (ii) the use of a lens system (L3 and L4) [43] to compensate for diffraction. In other words, by imaging the "output plane" at S3 to the "input plane" in the entrance of the DP. Finally, the optical power loss due to the filtering by the slits, might be addressed by using a pair of cylindrical lenses which can reduce each light stripe to a pair of spots. Alternatively, since the scheme uses classical states of light, losses can in principle be compensated by amplification inside the cavity.

The scalability of the scheme depends essentially on the ability to create and image variations of the transversal electromagnetic field on arbitrarily small length scales. However, it does not depend on the ability to detect these variations since only the integral over the electric field is measured. Metaphorically speaking, variation is increased with our method by cutting and pasting parts of the electromagnetic field in the transversal plane. Such increase of the spatial variation is accompanied by a growth of larger transversal wave vector components k_x in the Fourier spectrum. As far as this growth generates diverging light this is not a problem and can be controlled by lenses L3 and L4 that image the output plane to the input plane. For monochromatic light of wavelength λ, further increase of k_x components beyond $k_x^2 = (2\pi/\lambda)^2 - k_y^2$ leads to an imaginary k_z component, given by $k_z = i\sqrt{k_x^2 + k_y^2 - k^2}$, i.e., a light field that stops propagating in its initial direction z after an exponential decay of its amplitude. Whether techniques from plasmonics to image such evanescent fields [44,45] can be used successfully in order to achieve scalability of our approach is the subject of future research. In the absence of such solutions, the scalability rests on the ability to run the scheme with light sources and optics for arbitrary small wavelengths λ. While we cannot see a principle reason why this should not be possible, it is certainly not feasible with current technology.

In this work, we proposed a new way to implement the Deutsch–Jozsa Algorithm by using classical light. Our scheme makes use of linear optical elements inside a ring cavity to efficiently solve the task. We are able to create 2^n states in n steps while encoding the functional values in the phase of the field. An experimental realisation of the scheme will give an indication of the extent of the domain of functions that can be tested. If the problem of losses can be solved satisfactorily, we expect to exceed the capacity of the Turing machines available at the time of publishing the present results.

Acknowledgements

We are grateful for discussions to R. Simon and K. Mpofu. This work is based on the research supported by the National Research Foundation Grant 93102. B.P. and R.I.H. acknowledge support from the Consejo Nacional de Ciencia y Tecnología (grant 158174). S.K.G. would like would like to acknowledge support from NSERC.

Appendix A. Fourier transform

The definition of the two-dimensional Fourier transform of the complex function $g(x, y)$ reads

$$\mathcal{F}\{g(x,y)\} = \iint_{-\infty}^{\infty} g(x,y) \exp[-i2\pi(ux + vy)]\,\mathrm{d}x\,\mathrm{d}y, \tag{A.1}$$

where u and v are generally referred as the frequencies in Fourier space. Notice that if we evaluate at $u = v = 0$, the Fourier transform is equivalent to

$$\mathcal{F}\{g(x,y)\} = \iint_{-\infty}^{\infty} g(x,y)\,\mathrm{d}x\,\mathrm{d}y, \tag{A.2}$$

which is the integral of the function $g(x, y)$ over the x–y plane. Furthermore, it is well known that a lens can efficiently perform a two dimensional Fourier transform [41]. This means that using a lens and measuring at the origin ($u = v = 0$) defined by the optical axis in the focal plane behind the lens, we obtain the integral of the field entering the lens from the focal plane in front of the lens.

Appendix B. Finite length of 2^n states

By adapting the orientation of the Dove prism in the setup, the number of light spots can be doubled in such a way that the minimal distance is halved, starting with initial distance d and leading to total length L of

$$L = 2^{n+1} d/2^n = 2d \tag{B.1}$$

which is constant in all round trips $n = 1, 2 \ldots$ The condition for the rotation of the square pattern of spots in Fig. 5(c) to Fig. 5(d) is given in terms of the rotation angle ϕ about an axis through the centre of the pattern:

$$\tan\phi = \frac{d}{d/2^n} = \frac{1}{2^n}\,. \tag{B.2}$$

The Dove prism must thus be oriented by the angle $\phi/2$ with respect to its longitudinal axis.

If we consider the finite size of the spots we have the following relation

$$n = 1 + \log_2(d/\delta), \tag{B.3}$$

where n is the iteration inside the cavity, d is the initial separation between spots and δ is the size of the dots.

Appendix C. Identification of encoded function for partially overlapping spots

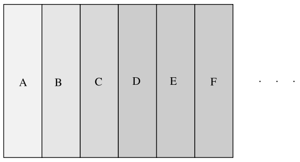

Fig. 6. Schematic representation of overlaps of light dots. Section A only carries the electric field of the left-most dot corresponding to the functional value $f(x_0)$. Section B contains a superposition of electric fields from the first two dots on the left corresponding to the sum of the functional values $f(x_0) + f(x_1)$. Other sections: $C = f(x_0) + f(x_1) + f(x_2)$, $D = f(x_0) + f(x_1) + f(x_2) + f(x_3)$, $E = f(x_1) + f(x_2) + f(x_3) + f(x_4)$ and $F = f(x_2) + f(x_3) + f(x_4) + f(x_5)$ etc.. The information about all functional values is thus encoded in the light field.

Light fields generated in our scheme with partial overlap between dots still carry the full information about the function. In order to see this, it suffices to look only at one of the two rows of dots. Neither the realisation scheme of the Deutsch–Jozsa algorithm nor the generalised Deutsch–Jozsa algorithm generates any two dots that perfectly overlap. Each row contains one dot at the end of the row that has only a neighbouring dot on one of its sides. In case it is on the right, the left section of the dot carries the electric field that corresponds to the functional value $f(x_0)$ for a particular argument x_0 (cp. Fig. 6). In the second section, there is a superposition of two electric fields corresponding to $f(x_0) + f(x_1)$, in the third section the electric field value corresponds to $f(x_0) + f(x_1) + f(x_2)$ and so forth. Thus the single functional values are represented in the electric field and could be identified, for example in the cases above by subtracting the functional value of the section on the left.

Appendix D. Partial overlap does not change readout

For the implementation of the Deutsch–Jozsa algorithm the programming leads to a faithful representation of the function despite overlap. Since in our implementations the functional values $f(x)$ are encoded as +1 and −1, a possible phase shift applied to the complete lower row changes the value $\sum_i f(x_i)$ of any given sector (cp. Fig. 6) to $\sum_i(-f(x_i))$. This leads to a change of the sign of each value $f(x_i) \rightarrow -f(x_i)$ that can be identified according to the recipe given in Appendix C. Moreover, the action of the pair of cylindrical lenses L1 and L2 in conjunction with the double slit S3 in Fig. 4 produces equal copies of light dots in the two rows

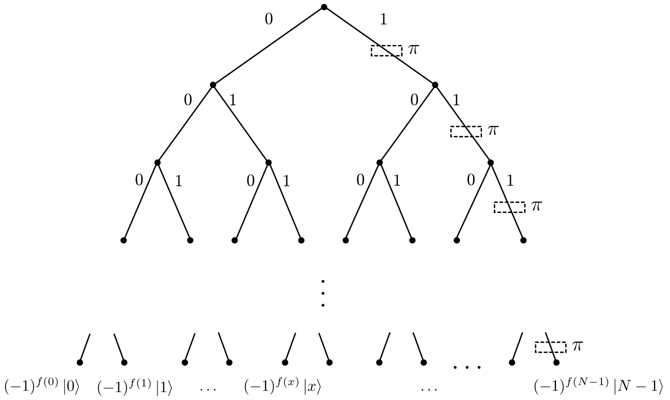

Fig. 7. Particular functions f with $N = 2^n$ arguments x assuming any number $0 \leq M \leq N$ of "1" values can be encoded by choosing to apply π phase shifts only in the right-most branches.

containing the full information about the function. Therefore the electric fields in the lower row either sums to zero (in case one or more phase plates have already been applied to the lower row in previous round trips) or (otherwise) the electric fields sum to a value corresponding to the full intensity, characteristic for a constant function. In the first case, the application of a phase shift to the light in the lower row will leave the sum of electric fields to be zero and in the second case such a phase shift will result in the following round trip into a zero total sum of electric fields. Consequently balanced functions will in the presence of overlap result in a detection of zero intensity, while constant functions will show full intensity – just as without overlap.

Appendix E. Generalised Deutsch–Jozsa problem

The Generalised Deutsch–Jozsa algorithm (GDJ) can be understood as follows: suppose we have a boolean function $f(x)$: $\{0,1\}^n \rightarrow \{0,1\}$ encoded in an oracle, such that the function possesses an arbitrary number of ones and zeros regardless of the order. The task is to find the number of ones of the function $f(x)$. The quantum circuit from the typical Deutsch–Jozsa algorithm can be used for the generalised version, leaving the final state as Eq. (3)

$$|\psi_{\text{out}}\rangle = \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{z=0}^{2^n-1}(-1)^{x\cdot z+f(x)}\,|z\rangle\,.$$

Please note, that in a classical Turing machine we need as many calls to the oracle as input values; nonetheless in a quantum approach the interference effects relate the probability of the state $|z\rangle = |0,0,\ldots,0\rangle$ with the number of ones encoded in $f(x)$.

A binary tree representation of the generalised Deutsch–Jozsa implementation is sketched in Fig. 7. It shows a way to encode for each number $0 \leq M \leq N$ an unbalanced function with M functional values equal to one. This task can be added to the scheme proposed in Sec. 4 with the restriction that one light dot must be addressable individually in any round trip, and thus during the encoding is kept separate from the other light dots.

The experimental setup for the generalised Deutsch–Jozsa is depicted in Fig. 8. In contrast to the Deutsch–Jozsa scheme, instead of a Dove prism, it uses a glass plate (GP) in the lower row to perform a lateral shift in the n-th round trip about the distance $\delta + d_{n-1}/2$ where δ is the dot width and d_{n-1} the previous distance between dots, see Fig. 9. This selection allows us to maintain the light dot on the right of the lower row with no overlap and apply to it individual phase shifts.

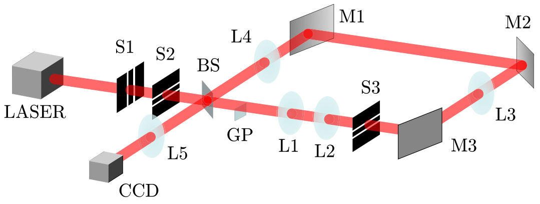

Fig. 8. Experimental implementation. S1–S3: slits; BS: beam splitter; GP: glass plate; L1 and L2 cylindrical lenses; L3–L5 Fourier transforming lens; M1–M3: mirrors.

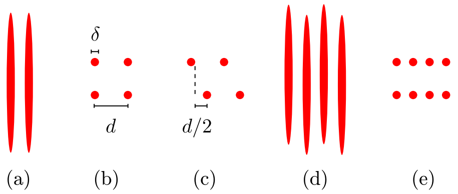

Fig. 9. Output after: (a) S1, (b) S2, (c) GP, (d) L2, (e) S3.

References

[1] D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, Proc. R. Soc. A 400 (1818) (1985) 97–117, http://dx.doi.org/10.1098/rspa.1985.0070.

[2] D. Deutsch, R. Jozsa, Rapid solution of problems by quantum computation, Proc. R. Soc. A 439 (1907) (1992) 553–558, http://dx.doi.org/10.1098/rspa.1992.0167.

[3] J.A. Jones, M. Mosca, Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer, J. Chem. Phys. 109 (5) (1998) 1648, http://dx.doi.org/10.1063/1.476739.

[4] N. Linden, H. Barjat, R. Freeman, An implementation of the Deutsch–Jozsa algorithm on a three-qubit NMR quantum computer, Chem. Phys. Lett. 296 (1–2) (1998) 61–67, http://dx.doi.org/10.1016/s0009-2614(98)01015-x.

[5] K. Dorai Arvind, A. Kumar, Implementing quantum-logic operations, pseudopure states, and the Deutsch–Jozsa algorithm using noncommuting selective pulses in NMR, Phys. Rev. A 61 (4) (2000), http://dx.doi.org/10.1103/physreva.61.042306.

[6] J. Kim, J.-S. Lee, S. Lee, C. Cheong, Implementation of the refined Deutsch–Jozsa algorithm on athree-bit NMR quantum computer, Phys. Rev. A 62 (2) (2000), http://dx.doi.org/10.1103/physreva.62.022312.

[7] T. Mahesh, K. Dorai Arvind, A. Kumar, Implementing logic gates and the Deutsch–Jozsa quantum algorithm by two-dimensional NMR using spin- and transition-selective pulses, J. Magn. Reson. 148 (1) (2001) 95–103, http://dx.doi.org/10.1006/jmre.2000.2225.

[8] O. Mangold, A. Heidebrecht, M. Mehring, NMR tomography of the three-qubit Deutsch–Jozsa algorithm, Phys. Rev. A 70 (4) (2004), http://dx.doi.org/10.1103/physreva.70.042307.

[9] S.-B. Zheng, Scheme for implementing the Deutsch–Jozsa algorithm in cavity QED, Phys. Rev. A 70 (3) (2004), http://dx.doi.org/10.1103/physreva.70.034301.

[10] Y. Rong-Can, L. Hong-Cai, L. Xiu, C. Mei-Xiang, Implementing the Deutsch–Jozsa algorithm by using Schrödinger cat states in cavity QED, Chin. Phys. 15 (10) (2006) 2320–2323, http://dx.doi.org/10.1088/1009-1963/15/10/021.

[11] W. Hong-Fu, Z. Shou, Implementation of n-qubit Deutsch–Jozsa algorithm using resonant interaction in cavity QED, Chin. Phys. B 17 (4) (2008) 1165–1173, http://dx.doi.org/10.1088/1674-1056/17/4/003.

[12] H.-F. Wang, S. Zhang, Y.-F. Zhao, Implementing Deutsch–Jozsa algorithm with superconducting quantum interference devices in cavity QED, Int. J. Theor. Phys. 48 (8) (2009) 2384–2389, http://dx.doi.org/10.1007/s10773-009-0028-8.

[13] P. Bianucci, A. Muller, C.K. Shih, Q.Q. Wang, Q.K. Xue, C. Piermarocchi, Experimental realization of the one qubit Deutsch–Jozsa algorithm in a quantum dot, Phys. Rev. B 69 (16) (2004), http://dx.doi.org/10.1103/physrevb.69.161303.

[14] M. Scholz, T. Aichele, S. Ramelow, O. Benson, Deutsch–Jozsa algorithm using triggered single photons from a single quantum dot, Phys. Rev. Lett. 96 (18) (2006), http://dx.doi.org/10.1103/physrevlett.96.180501.

[15] S. Gulde, M. Riebe, G.P.T. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I.L. Chuang, R. Blatt, Implementation of the Deutsch–Jozsa algorithm on an ion-trap quantum computer, Nature 421 (6918) (2003) 48–50, http://dx.doi.org/10.1038/nature01336.

[16] S. Dasgupta, A. Biswas, G.S. Agarwal, Implementing Deutsch–Jozsa algorithm using light shifts and atomic ensembles, Phys. Rev. A 71 (1) (2005), http://dx.doi.org/10.1103/physreva.71.012333.

[17] X.-H. Zheng, M. Yang, P. Dong, Z.-L. Cao, Implementing Deutsch–Jozsa algorithm using superconducting qubit network, Mod. Phys. Lett. B 22 (31) (2008) 3035–3042, http://dx.doi.org/10.1142/s0217984908017540.

[18] F. Shi, X. Rong, N. Xu, Y. Wang, J. Wu, B. Chong, X. Peng, J. Kniepert, R.-S. Schoenfeld, W. Harneit, et al., Room-temperature implementation of the Deutsch–Jozsa algorithm with a single electronic spin in diamond, Phys. Rev. Lett. 105 (4) (2010), http://dx.doi.org/10.1103/physrevlett.105.040504.

[19] A.N.d. Oliveira, S.P. Walborn, C.H. Monken, Implementing the Deutsch algorithm with polarization and transverse spatial modes, J. Opt. B, Quantum Semiclass. Opt. 7 (9) (2005) 288–292, http://dx.doi.org/10.1088/1464-4266/7/9/009.

[20] B. Marques, M.R. Barros, W.M. Pimenta, M.A.D. Carvalho, J. Ferraz, R.C. Drumond, M. Terra Cunha, S. Pádua, Double-slit implementation of the minimal Deutsch algorithm, Phys. Rev. A 86 (3) (2012), http://dx.doi.org/10.1103/physreva.86.032306.

[21] P. Zhang, R.-F. Liu, Y.-F. Huang, H. Gao, F.-L. Li, Demonstration of Deutsch's algorithm on a stable linear optical quantum computer, Phys. Rev. A 82 (6) (2010), http://dx.doi.org/10.1103/physreva.82.064302.

[22] P. Zhang, Y. Jiang, R.-F. Liu, H. Gao, H.-R. Li, F.-L. Li, Implementing the Deutsch's algorithm with spin-orbital angular momentum of photon without interferometer, Opt. Commun. 285 (5) (2012) 838–841, http://dx.doi.org/10.1016/j.optcom.2011.11.024.

[23] N. Schuch, J. Siewert, Implementation of the four-bit Deutsch–Jozsa algorithm with Josephson charge qubits, Phys. Status Solidi B 233 (3) (2002) 482–489, http://dx.doi.org/10.1002/1521-3951(200210)233:3<482::AID-PSSB482>3.0.CO.

[24] B. Perez-Garcia, J. Francis, M. McLaren, R.I. Hernandez-Aranda, A. Forbes, T. Konrad, Quantum computation with classical light: the Deutsch algorithm, Phys. Lett. A 379 (28–29) (2015) 1675–1680, http://dx.doi.org/10.1016/j.physleta.2015.04.034.

[25] N. Johansson, J.-Å. Larsson, Efficient algorithms for the Deutsch–Jozsa and Simon's problems that only use classical resources, arXiv:1508.05027.

[26] D. Collins, K.W. Kim, W.C. Holton, Deutsch–Jozsa algorithm as a test of quantum computation, Phys. Rev. A 58 (3) (1998) R1633–R1636, http://dx.doi.org/10.1103/physreva.58.r1633.

[27] G. Puentes, C.L. Mela, S. Ledesma, C. Iemmi, J.P. Paz, M. Saraceno, Optical simulation of quantum algorithms using programmable liquid-crystal displays, Phys. Rev. A 69 (2004) 042319, http://dx.doi.org/10.1103/PhysRevA.69.042319.

[28] M. Nielsen, I. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge and New York, 2000.

[29] R. Raussendorf, H.J. Briegel, A one-way quantum computer, Phys. Rev. Lett. 86 (2001) 5188–5191, http://dx.doi.org/10.1103/PhysRevLett.86.5188.

[30] E. Farhi, S. Gutmann, Quantum computation and decision trees, Phys. Rev. A 58 (1998) 915–928, http://dx.doi.org/10.1103/PhysRevA.58.915.

[31] S. Kotiyal, H. Thapliyal, N. Ranganathan, Circuit for reversible quantum multiplier based on binary tree optimizing Ancilla and Garbage bits, in: 27th International Conference on VLSI Design and 13th International Conference on Embedded Systems, 2014, pp. 545–550.

[32] D. Dragoman, n-step optical simulation of the n-qubit state: applications in optical computing, Optik 113 (10) (2002) 425–428, http://dx.doi.org/10.1078/s0030-4026(04)70193-x.

[33] R.J.C. Spreeuw, A classical analogy of entanglement, Found. Phys. 28 (3) (1998) 361–374, http://dx.doi.org/10.1023/a:1018703709245.

[34] A. Luis, Coherence, polarization, and entanglement for classical light fields, Opt. Commun. 282 (18) (2009) 3665–3670, http://dx.doi.org/10.1016/j.optcom.2009.06.024.

[35] B.N. Simon, S. Simon, F. Gori, M. Santarsiero, R. Borghi, N. Mukunda, R. Simon, Nonquantum entanglement resolves a basic issue in polarization optics, Phys. Rev. Lett. 104 (2010) 023901, http://dx.doi.org/10.1103/PhysRevLett.104.023901.

[36] M. McLaren, T. Konrad, A. Forbes, Measuring the non-separability of classically entangled vector vortex beams, Phys. Rev. A 92 (2) (2015) 023833, http://dx.doi.org/http://dx.doi.org/10.1103/PhysRevA.92.023833.

[37] A. Aiello, F. Töppel, C. Marquardt, E. Giacobino, G. Leuchs, Classical entanglement: oxymoron or resource?, arXiv:1409.0213, 2014.

[38] S. Goyal, F. Roux, A. Forbes, T. Konrad, Implementing quantum walks using orbital angular momentum of classical light, Phys. Rev. Lett. 110 (2013) 263602.

[39] S.K. Goyal, F.S. Roux, A. Forbes, T. Konrad, Implementation of multidimensional quantum walks using linear optics and classical light, Phys. Rev. A 92 (4) (2015) 040302, http://dx.doi.org/10.1103/PhysRevA.92.040302.

[40] D. Tamir, N. Shaked, P. Wilson, S. Dolev, High-speed and low-power electro-optical DSP coprocessor, J. Opt. Soc. Am. A 26 (2009) A11–A20.

[41] J. Goodman, Introduction to Fourier Optics, McGraw–Hill Series in Electrical and Computer Engineering: Communications and Signal Processing, McGraw–Hill, 1996.

[42] R.J.C. Spreeuw, Classical wave-optics analogy of quantum-information processing, Phys. Rev. A 63 (2001) 062302, http://dx.doi.org/10.1103/PhysRevA.63.062302.

[43] E. Sudarshan, N. Mukunda, R. Simon, Realization of first order optical systems using thin lenses, Opt. Acta 32 (1985) 855–872.

[44] J.B. Pendry, Negative refraction makes a perfect lens, Phys. Rev. Lett. 85 (2000) 3966.

[45] Z. Zhang, P. Ahn, B. Dong, O. Balogun, C. Sun, Quantitative imaging of rapidly decaying evanescent fields using plasmonic near-field scanning optical microscopy, Sci. Rep. 3 (2013) 2803.