A quantum probability splitter and its application to qubit preparation

Kyriakos N. Sgarbas

Received: 17 August 2011 / Accepted: 27 March 2012 / Published online: 18 April 2012 © The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract This paper presents a quantum probability splitter, i.e. a quantum circuit that modifies the probability amplitudes of a qubit so that the probability on the selected basis state is halved. It also presents a potential application of this circuit related to qubit preparation: it shows how a qubit is prepared to a superposition of any valid pair of basis states probabilities, by repeated application of the quantum probability splitter.

Keywords Quantum circuit · Quantum probability · Qubit preparation · Quantum algorithms · Probability splitter · Probability halver

1 Introduction

The first and crucial [3] implementation stage in every quantum algorithm is to properly prepare the qubits of the quantum register. Although most current quantum algorithms (e.g. [2,5,9]) start with the register qubits in the basis states and prepare them with Hadamard gates [8], it would be useful to have a gate (or a process) that arbitrarily sets a qubit to a superposition state with any valid probability pair for its basis states. Indeed, theory suggests that this task is trivial (e.g. [4]) but this is true only if the underlying hardware allows the construction of arbitrary quantum gates on demand (especially arbitrary phase shift gates) to use in each specific calculation. And this is very inconvenient (or even impossible) for certain implementation schemes, such as solid-state quantum computers [1,7,10]. So, in cases where the implementation of the

K. N. Sgarbas (🖂)

Wire Communications Lab, Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece e-mail: sgarbas@upatras.gr

URL: http://www.wcl.ece.upatras.gr/sgarbas



precise quantum gate that would immediately set the qubit to the desired state is not feasible, it would prove useful to have a fixed quantum circuit that if used repeatedly will be able to set the qubit to at least a good approximation of the desired state.

This paper presents a quantum circuit that redistributes the probabilities between the basis states of a qubit proportionally. In fact, the circuit allows the selection of either basis state of the initial superposition and halves its probability, transferring the other half to the other basis state.

This idea is shown schematically in Fig. 1 where two jars (labeled $|0\rangle$ and $|1\rangle$) initially contain half a liter of liquid each. Only one operation is allowed: "*Pick a jar and pour half of its content to the other.*" If we call this splitting operation A, then every time operation A is applied the sum of the contents of the jars remains constant (equal to 1), but it is possible to approximate any amount x(0 < x < 1) of content in any of the jars with arbitrary precision¹ by repeated A operations, i.e. by successively halving the content of one or the other jar (see Sect. 4 for the exact process). A quantum gate with this behavior could prove handy in quantum algorithms.

The construction of a quantum circuit performing the aforementioned probability splitting operation is explained in Sect. 2. Section 3 verifies the circuit's function and lists its outputs in several diverse inputs. Section 4 explains how to use the quantum probability splitter to set a qubit to a desired state with arbitrary precision, and Sect. 5 concludes the paper with an overview and some final remarks.

2 Constructing a quantum probability splitter

A (hypothetical) quantum gate that behaves like the aforementioned probability splitter is shown in Fig. 2, along with its intended truth table.

Qubit $|q\rangle$ is the target qubit, while $|s\rangle$ functions as a switch: it determines which of the basis states of $|q\rangle$ to pick and halve its probability. So, in the inputs $|01\rangle$ and $|10\rangle$

¹ This problem arises in many scientific areas, e.g. in artificial intelligence (as a generalization of the famous water jug problem, where the search space is continuous), in fractal geometry (where by splitting a line in half recursively we can approximate any point of it), in the theory of algorithms (a binary search process can reach any element of a sorted array), in binary arithmetic (where any real number can be approximated by a sum of powers of 2) etc.



where the selected probability is zero, the output remains the same, but in the other cases the target qubit $|q\rangle$ is set to the state $|+\rangle$, causing the probability of the selected component to split in two.

If $|q\rangle = a |0\rangle + b |1\rangle$, with $P_q(0) = a^2$ and $P_q(1) = b^2$, then the splitter has the following effect on the qubit:

$$A |sq\rangle =_{for s=0} A |0\rangle (a |0\rangle + b |1\rangle) = aA |00\rangle + bA |01\rangle = a |0+\rangle + b |01\rangle$$
$$= a |0\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + b |01\rangle = \frac{a}{\sqrt{2}} |00\rangle$$
$$+ \frac{a}{\sqrt{2}} |01\rangle + b |01\rangle = \frac{a}{\sqrt{2}} |00\rangle$$
$$+ \left(\frac{a}{\sqrt{2}} + b\right) |01\rangle$$
(1)

And although $P_{q'}(0) = \frac{a^2}{2} = \frac{P_q(0)}{2}$ as intended, it also holds that $P_{q'}(0) + P_{q'}(1) = \frac{a^2}{2} + \left(\frac{a}{\sqrt{2}} + b\right)^2 = a^2 + b^2 + \sqrt{2}ab > 1$, which implies that gate A cannot exist. Indeed, if it existed then its 4x4 matrix should be:

$$A = \begin{bmatrix} \langle 00|0+\rangle & \langle 00|01\rangle & \langle 00|10\rangle & \langle 00|1+\rangle \\ \langle 01|0+\rangle & \langle 01|01\rangle & \langle 01|10\rangle & \langle 01|1+\rangle \\ \langle 10|0+\rangle & \langle 10|01\rangle & \langle 10|10\rangle & \langle 10|1+\rangle \\ \langle 11|0+\rangle & \langle 11|01\rangle & \langle 11|10\rangle & \langle 11|1+\rangle \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$
(2)

And this matrix is not unitary.

At Eq. 1 before we perform the final addition $\frac{a}{\sqrt{2}} |00\rangle + \frac{a}{\sqrt{2}} |01\rangle + b |01\rangle$, the probabilities could sum to $a^2 + b^2 = 1$, if the two $|01\rangle$ terms were distinct. This suggests the introduction of an extra (ancilla) qubit as shown in Fig. 3. The first two qubits in Fig. 3 have the same functions as the ones in Fig. 2. The last one is the ancilla qubit. It is set to $|0\rangle$ before the operation and its output state is marked as $|d\rangle$ (don't care).

The probability splitter A can be implemented with the quantum circuit shown in Fig. 4. The logic behind its construction is the following: using the intended truth

 $\begin{array}{c|c} |00\rangle & \stackrel{A}{\rightarrow} & |0+\rangle \\ |01\rangle & \stackrel{A}{\rightarrow} & |01\rangle \end{array}$

 $|s\rangle$

² In order to simplify the notation throughout the paper the amplitudes *a* and *b* are considered real numbers and the probabilities are expressed as a^2 and b^2 instead of $|a|^2$ and $|b|^2$. The effect of the circuit on target qubits with phases is explained in Sect. 3.



Fig. 3 Schematic and truth table of a functional 3-qubit quantum probability splitter. The lower qubit is ancilla



Fig. 4 A quantum circuit for the quantum probability splitter A of Fig. 3 with a full truth table

table (Figs. 2 and 3) as a guide, with the ancilla qubit set to $|0\rangle$, the first cNOT-NOT sequence checks if $|s\rangle$ and $|q\rangle$ are equal. If this happens $|q\rangle$ is set to $|1\rangle$ at point P₂ and the controlled-Hadamard (cH) changes the ancilla qubit to $|+\rangle$ at point P₃. Otherwise, the ancilla qubit remains $|0\rangle$. Now, for every one of the four basis states for $|sq\rangle$ in the truth table, the output after cH (point P₃) is almost ready, except for the case $|sq\rangle = |01\rangle$, where it outputs $|0d0\rangle$ instead of $|0d1\rangle$. Thus another unitary transformation is used to exchange $|000\rangle$ to $|001\rangle$ and vice-versa, leaving all other basis states unaffected. This transformation can be constructed using a negated Toffoli gate [6], here depicted as a Toffoli with four NOT gates between P₃ and P₄. The last gate exchanges the 2nd and 3rd qubits just to move the ancilla out of the way.

A single matrix representation for the whole circuit can be calculated as:

One can easily verify that this matrix is unitary, since $AA^{\dagger} = A^{\dagger}A = I$.

The circuit of Fig. 4 can be simplified in numerous ways: the two NOT gates after the Toffoli can be omitted, since they affect only $|s\rangle$ and $|d\rangle$. Also, if the order of the



Fig. 5 A simplified version of the quantum probability splitter circuit

qubits in the output is not important, then the Exchange gate can be removed as well. Further simplification can be achieved by removing the NOT gate between P₁ and P₂, only this way the function of $|s\rangle$ is reversed (the value $|0\rangle$ would mean "halve the probability of state $|1\rangle$)" and consequently the NOT gates between P₃ and P₄, should be changed accordingly. Figure 5 shows the result of all these simplifications.³ The simplified circuit has much less gates and it is functionally equivalent to the original of Fig. 4. However, it conforms to a different truth table that reverses the function of $|s\rangle$ and does not retain the order of the last two qubits. Thus it cannot directly replace the original in the application of Sect. 4. The analysis in this paper is based on the original circuit of Fig. 4 as the most general representative of all the circuits corresponding to the aforementioned simplifications.

3 Verifying the behavior of the quantum probability splitter for various inputs

In order to verify the function of operator A, its effect on the generic input state $|q\rangle = a |0\rangle + b |1\rangle$ can be calculated algebraically. For instance the application of A to the input vector $|0q0\rangle$ is calculated as:

$$0q0\rangle = |0\rangle (a |0\rangle + b |1\rangle) |0\rangle = a |000\rangle + b |010\rangle \xrightarrow{A} a |0+1\rangle + b |010\rangle$$
$$= a |0\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) |1\rangle + b |010\rangle = \frac{a}{\sqrt{2}} |001\rangle + \frac{a}{\sqrt{2}} |011\rangle + b |010\rangle \quad (4)$$

And the probabilities for the target (middle) qubit are: $P_{q'}(0) = \frac{a^2}{2}, P_{q'}(1) = \frac{a^2}{2} + b^2$.

Respectively, the application of A to the input vector $|1q0\rangle$ derives $a |100\rangle + \frac{b}{\sqrt{2}} |101\rangle + \frac{b}{\sqrt{2}} |111\rangle$ and the probabilities are: $P_{q'}(0) = a^2 + \frac{b^2}{2}$, $P_{q'}(1) = \frac{b^2}{2}$. Thus is verified that the application of A to a target qubit halves the probability of

Thus is verified that the application of A to a target qubit halves the probability of the selected basis component and transfers the other half to the other component.

Although the above two input states $(|0q0\rangle \text{ and } |1q0\rangle)$ are the most useful for controlling the probability distribution, Table 1 lists the results of the application of operator A to several other inputs as well, including superpositions of $|s\rangle$ and different initial values for the ancilla qubit. It is interesting that setting the ancilla qubit to $|1\rangle$ causes the probabilities to switch basis states after the split. Even more interesting is

³ Note that according to the actual implementation, other types of simplification might also apply (e.g. a single negated Toffoli [6] between P_3 and P_4).

Input vector		Output vector	P(0)	P(1)
$ 0q0\rangle$	$\begin{bmatrix} a & 0 & b & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$	$\left[\begin{array}{ccc} 0 & \frac{a}{\sqrt{2}} & b & \frac{a}{\sqrt{2}} & 0 & 0 & 0 \\ \end{array} \right]^T$	$\frac{a^2}{2}$	$\frac{a^2}{2} + b^2$
$ 1q0\rangle$	$\begin{bmatrix} 0 & 0 & 0 & 0 & a & 0 & b & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 0 & 0 & 0 & a & \frac{b}{\sqrt{2}} & 0 & \frac{b}{\sqrt{2}} \end{bmatrix}^T$	$a^2 + \frac{b^2}{2}$	$\frac{b^2}{2}$
$ +q0\rangle$	$\left[\frac{a}{\sqrt{2}} \ 0 \ \frac{b}{\sqrt{2}} \ 0 \ \frac{a}{\sqrt{2}} \ 0 \ \frac{b}{\sqrt{2}} \ 0 \right]^T$	$\begin{bmatrix} 0 & \frac{a}{2} & \frac{b}{\sqrt{2}} & \frac{a}{2} & \frac{a}{\sqrt{2}} & \frac{b}{2} & 0 & \frac{b}{2} \end{bmatrix}^T$	$\frac{3a^2}{4} + \frac{b^2}{4}$	$\frac{a^2}{4} + \frac{3b^2}{4}$
$ -q0\rangle$	$\begin{bmatrix} \frac{a}{\sqrt{2}} & 0 & \frac{b}{\sqrt{2}} & 0 & \frac{-a}{\sqrt{2}} & 0 & \frac{-b}{\sqrt{2}} & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & \frac{a}{2} & \frac{b}{\sqrt{2}} & \frac{a}{2} & \frac{-a}{\sqrt{2}} & \frac{-b}{2} & 0 & \frac{-b}{2} \end{bmatrix}^T$	$\frac{3a^2}{4} + \frac{b^2}{4}$	$\frac{a^2}{4} + \frac{3b^2}{4}$
$ 0q1\rangle$	$\begin{bmatrix} 0 & a & 0 & b & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} b & \frac{a}{\sqrt{2}} & 0 & \frac{-a}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{bmatrix}^T$	$\frac{a^2}{2} + b^2$	$\frac{a^2}{2}$
$ 1q1\rangle$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & a & 0 & b \end{bmatrix}^T$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{b}{\sqrt{2}} & a & \frac{-b}{\sqrt{2}} \end{bmatrix}^T$	$\frac{b^2}{2}$	$a^2 + \frac{b^2}{2}$
$ +q1\rangle$	$\left[\begin{array}{ccc} 0 & \frac{a}{\sqrt{2}} & 0 & \frac{b}{\sqrt{2}} & 0 & \frac{a}{\sqrt{2}} & 0 & \frac{b}{\sqrt{2}} \end{array} \right]^T$	$\begin{bmatrix} \frac{b}{\sqrt{2}} & \frac{a}{2} & 0 & \frac{-a}{2} & 0 & \frac{b}{2} & \frac{a}{\sqrt{2}} & \frac{-b}{2} \end{bmatrix}^T$	$\frac{a^2}{4} + \frac{3b^2}{4}$	$\frac{3a^2}{4} + \frac{b^2}{4}$
$ -q1\rangle$	$\begin{bmatrix} 0 & \frac{a}{\sqrt{2}} & 0 & \frac{b}{\sqrt{2}} & 0 & \frac{-a}{\sqrt{2}} & 0 & \frac{-b}{\sqrt{2}} \end{bmatrix}^T$	$\begin{bmatrix} \frac{b}{\sqrt{2}} & \frac{a}{2} & 0 & \frac{-a}{2} & 0 & \frac{-b}{2} & \frac{-a}{\sqrt{2}} & \frac{b}{2} \end{bmatrix}^T$	$\frac{a^2}{4} + \frac{3b^2}{4}$	$\frac{3a^2}{4} + \frac{b^2}{4}$
$ 0q+\rangle$	$\begin{bmatrix} \frac{a}{\sqrt{2}} & \frac{a}{\sqrt{2}} & \frac{b}{\sqrt{2}} & \frac{b}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} \frac{b}{\sqrt{2}} & a & \frac{b}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{bmatrix}^T$	$a^2 + \frac{b^2}{2}$	$\frac{b^2}{2}$
$ 1q+\rangle$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{a}{\sqrt{2}} & \frac{a}{\sqrt{2}} & \frac{b}{\sqrt{2}} & \frac{b}{\sqrt{2}} \end{bmatrix}^T$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{a}{\sqrt{2}} & b & \frac{a}{\sqrt{2}} & 0 \end{bmatrix}^T$	$\frac{a^2}{2} + b^2$	$\frac{a^2}{2}$
$ +q+\rangle$	$\begin{bmatrix} \underline{a} & \underline{a} & \underline{b} & \underline{b} & \underline{a} & \underline{a} & \underline{b} & \underline{b} \\ \underline{2} & \underline{2} \end{bmatrix}^T$	$\begin{bmatrix} \frac{b}{2} & \frac{a}{\sqrt{2}} & \frac{b}{2} & 0 & \frac{a}{2} & \frac{b}{\sqrt{2}} & \frac{a}{2} & 0 \end{bmatrix}^T$	$\frac{3}{4}$	$\frac{1}{4}$
$ -q+\rangle$	$\begin{bmatrix} \underline{a} & \underline{a} & \underline{b} & \underline{b} & \underline{-a} & \underline{-a} & \underline{-b} & \underline{-b} \\ \underline{a} & \underline{b} & \underline{b} & \underline{b} & \underline{-a} & \underline{-b} & \underline{-b} \end{bmatrix}^T$	$\begin{bmatrix} \frac{b}{2} & \frac{a}{\sqrt{2}} & \frac{b}{2} & 0 & \frac{-a}{2} & \frac{-b}{\sqrt{2}} & \frac{-a}{2} & 0 \end{bmatrix}^T$	$\frac{3}{4}$	$\frac{1}{4}$
$ 0q-\rangle$	$\begin{bmatrix} \frac{a}{\sqrt{2}} & \frac{-a}{\sqrt{2}} & \frac{b}{\sqrt{2}} & \frac{-b}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix}^{T}$	$\left[\frac{-b}{\sqrt{2}} \ 0 \ \frac{b}{\sqrt{2}} \ a \ 0 \ 0 \ 0 \ 0 \right]^T$	$\frac{b^2}{2}$	$a^2 + \frac{b^2}{2}$
$ 1q-\rangle$	$\begin{bmatrix} 0 & 0 & 0 & \frac{a}{\sqrt{2}} & \frac{-a}{\sqrt{2}} & \frac{b}{\sqrt{2}} & \frac{-b}{\sqrt{2}} \end{bmatrix}^T$	$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{a}{\sqrt{2}} & 0 & \frac{-a}{\sqrt{2}} & b \end{bmatrix}^T$	$\frac{a^2}{2}$	$\frac{a^2}{2} + b^2$
$ +q-\rangle$	$\begin{bmatrix} \underline{a} & \underline{-a} & \underline{b} & \underline{-b} & \underline{a} & \underline{-a} & \underline{b} & \underline{-b} \\ \underline{2} & \underline{2} \end{bmatrix}^T$	$\left[\begin{array}{ccc} \frac{-b}{2} & 0 & \frac{b}{2} & \frac{a}{\sqrt{2}} & \frac{a}{2} & 0 & \frac{-a}{2} & \frac{b}{\sqrt{2}} \end{array}\right]^T$	$\frac{1}{4}$	$\frac{3}{4}$
$\left -q-\right\rangle$	$\begin{bmatrix} \underline{a} & \underline{-a} & \underline{b} & \underline{-b} & \underline{-a} & \underline{a} & \underline{-b} & \underline{b} \\ \underline{2} & \underline{2} \end{bmatrix}^T$	$\begin{bmatrix} \frac{-b}{2} & 0 & \frac{b}{2} & \frac{a}{\sqrt{2}} & \frac{-a}{2} & 0 & \frac{a}{2} & \frac{-b}{\sqrt{2}} \end{bmatrix}^T$	$\frac{1}{4}$	$\frac{3}{4}$

 Table 1
 A detailed record of the function of the quantum probability splitter circuit

the finding that for certain input configurations $(|+q+\rangle, |-q+\rangle, |+q-\rangle, |-q-\rangle)$ the probabilities are reset and distributed to 25% and 75% constantly, for any input qubit! Finally, if the input qubit has a phase, the application of A derives:

$$|0q0\rangle = |0\rangle \left(a |0\rangle + be^{i\phi} |1\rangle\right) |0\rangle = a |000\rangle + be^{i\phi} |010\rangle \xrightarrow{A} a |0+1\rangle + be^{i\phi} |010\rangle$$
$$= a |0\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) |1\rangle + be^{i\phi} |010\rangle = \frac{a}{\sqrt{2}} |001\rangle + \frac{a}{\sqrt{2}} |011\rangle + be^{i\phi} |010\rangle$$
(5)

$$|1q0\rangle = |1\rangle \left(a |0\rangle + be^{i\phi} |1\rangle \right) |0\rangle = a |100\rangle + be^{i\phi} |110\rangle \xrightarrow{A} a |001\rangle + be^{i\phi} |1+1\rangle$$
$$= a |100\rangle + be^{i\phi} |1\rangle \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) |1\rangle = a |100\rangle + \frac{be^{i\phi}}{\sqrt{2}} |101\rangle + \frac{be^{i\phi}}{\sqrt{2}} |111\rangle$$
(6)

Equations 5 and 6 verify that the phase is retained: every component that splits its probability carries along its initial phase as well.

For 16 combinations of the switch and ancilla qubits it displays the corresponding output vectors and the probabilities of observation P(0) and P(1) for the target qubit

It is not possible to predict exactly where a generic component such as the quantum probability splitter might be used. In this section one potential use is demonstrated, showing how to prepare a qubit to a state of desired probability distribution. By applying A successively to a qubit originally set to an initial basis vector as shown in Fig. 6a, the qubit can be prepared to a superposition state with any valid pair of amplitudes, with arbitrary precision, depending on the number of *iterations* (successive A operations). The process is controlled by setting appropriately the switch qubit s_i before each iteration. The diagram in Fig. 7 shows the exact probabilities obtained after each iteration, starting from the basis vectors. Every node in this diagram represents the probability pair (a^2, b^2) for the corresponding qubit state. Every transition represents the application of A with the switch value indicated on the label of the arrow. The states in Fig. 7 are grouped in levels (L = 0, 1, 2, 3, ...) according to how many

A operations away from the basis states they are. All probabilities in the same level L have the same denominator 2^{L} , which indicates that L also represents the resolution of the approximation in bits. Fig. 7 stops at L = 5, but the actual tree continues likewise infinitely.

Given a level limit L and a desired probability pair (p, 1 - p), Textbox 1 presents a process that calculates the number of required successive applications of A (less or equal to L) and the appropriate sequence of s_i values to best approximate the pair. In order to understand how the process works, note in the diagram of Fig. 7 that for every level L > 0, the numerators are all odd numbers between 0 and 2^L. In fact, every odd number appears twice, once in the left fraction of a pair (the white circles in Fig. 7), once in a right fraction. The binary representation of every left numerator (in the white circles) can be composed by the series of bit labels on the arrows, if the tree is traversed reversely from the numerator's state to (0, 1). The process of Textbox 1 has an O(L) complexity both in time and space, since the *while* loop and the *binary()* function



Fig. 6 a Successive application of operator A produces a qubit with probability amplitudes determined by the sequence of switches s_i . b The first application of A can always be replaced by a single Hadamard gate



Fig. 7 Successive application of A to a qubit initially prepared at a basis state, leads to states with rational probability pairs of increased resolution

require logarithmic time in respect to N and $N = p \cdot 2^{L}$. L is also the maximum length of *sequence[]*.

And since L also represents the resolution in bits, then the approximation error is at most $1/2^{L+1}$. This is a *quantization error*. In terms of *fidelity*, a lower bound can be calculated. If $|\psi\rangle$ is the intended state and $|\varphi\rangle$ the approximated one, then:

$$F\left(\rho_{\psi},\rho_{\varphi}\right) \ge \langle 1| \left(\frac{1}{\sqrt{2^{L+1}}} |0\rangle + \sqrt{\frac{2^{L+1}-1}{2^{L+1}}} |1\rangle\right) = \sqrt{1 - \frac{1}{2^{L+1}}}$$
(7)

Solving Eq. 7 for L we get the lowest L that ensures a certain degree of fidelity:

$$\mathcal{L} = ceiling\left(-\log_2\left(1 - F^2\right)\right) - 1 \tag{8}$$

Therefore, by applying A to (0, 1) as shown in Fig. 6a, a number of times determined by the process of Textbox 1, a qubit can be prepared to a superposition of any valid probability pair with fidelity at least as calculated by Eq. 7. The price for each iteration is not only the computational cost for applying matrix A, but also an additional ancilla qubit. Indeed in Fig. 6a the same switch qubit can be reused just by resetting it before a new iteration. But every ancilla qubit must be kept intact until $|q\rangle$ is measured, since:

```
INPUT: a probability pair (p, 1-p)
       a level limit L
                                 % the level limit also determines
                                 % the maximum number of iterations.
BEGIN
D ← 2^L
                          % D: the denominator.
N \leftarrow p*D
                          % N: the numerator; if N is integer,
                          % the probability pair can be matched exactly.
                          % nint(): the nearest integer function.
N \leftarrow nint(N)
while (N is even):
   N \leftarrow N/2
                          % Simplification of the fraction
    D \leftarrow D/2
                          % until the numerator becomes odd.
end while
B \leftarrow \log_2(D)
                                 % B: the number of digits (bits) of the
                                 % simplified denominator expressed in binary.
sequence[] \leftarrow binary(N,B)
                                 % sequence[]: a one dimensional array.
                                 % binary(N,B): a function converting
                                   a decimal integer N to an array of
                                 % size B of binary digits.
% the reversed array of X.
END
OUTPUT: sequence[]
                          \% the sequence of s_i.
        в
                          % B equals to the number of iterations and the
                          % length of sequence[].
```

Textbox 1 This algorithm outputs the number of iterations and the sequence of switch-values for approximating the given probability pair with successive applications of the quantum probability splitter A

$$|0q0\rangle \xrightarrow{A} \frac{a}{\sqrt{2}} |001\rangle + \frac{a}{\sqrt{2}} |011\rangle + b |010\rangle = |0\rangle \left(\frac{a}{\sqrt{2}} |01\rangle + \frac{a}{\sqrt{2}} |11\rangle + b |10\rangle\right)$$
$$|1q0\rangle \xrightarrow{A} a |100\rangle + \frac{b}{\sqrt{2}} |101\rangle + \frac{b}{\sqrt{2}} |111\rangle = |1\rangle \left(a |00\rangle + \frac{b}{\sqrt{2}} |01\rangle + \frac{b}{\sqrt{2}} |11\rangle\right)$$
(9)

Equations 9 show that qubit $|s\rangle$ is independent, but the other two are entangled.

However, we can always save an ancilla qubit (and an iteration) if we start the process from Level 1, state $(\frac{1}{2}, \frac{1}{2})$ instead of Level 0, state (0, 1), using a Hadamard gate as shown in Fig. 6b.

5 Conclusion

In this paper a quantum circuit was presented that affects the amplitudes of a target qubit so that the probability of a selected basis state is halved. It was proved that there cannot be a 2-qubit unitary operation able to perform this task. For the presented 3-qubit quantum probability splitter circuit, its operation was explained, its matrix representation was calculated and its unitarity was proved. Then its operation was verified and output vectors and probabilities were calculated for 16 different configurations of the switch and ancilla qubits.

As with most basic circuits, one cannot predict exactly where the quantum probability splitter could be applied in the future; it could be used to achieve a proportional probability redistribution of a qubit state mid-algorithmically, or its behavior in some of the special configurations of Table 1 might be handy in some other algorithm. In this paper it was demonstrated how the quantum probability splitter can be used to prepare the state of a qubit with arbitrary precision. Given the required fidelity, the value of L is calculated by Eq. 8 and the process of Textbox 1 returns the number of required iterations and the values s_i . For example, to ensure fidelity 0.9, 0.99, and 0.999, we need at most 2, 5 and 8 iterations respectively (1, 4, and 7 if we use the shortcut of Fig. 6b). The qubit preparation process was presented mostly as an example of use for the circuit, but it could also be handy in certain implementations of quantum computers.

A simplified version of the splitter was presented as well (Fig. 5). It is functionally equivalent to the main circuit of Fig. 4 (i.e. it splits probabilities as well), but it has a different truth table that does not facilitate its successive application as in Fig. 6. However, it uses considerably less gates than the original, thus it would be easier to implement and verify experimentally.

Acknowledgments The publication and open access costs for this article were covered by the University of Patras Research Committee (ELKE/TSMEDE fund, VAT: EL 998219694).

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Copsey, D., Oskin, M., Impens, F., Metodiev, T., Cross, A., Chong, F.T., Chuang, I.L., Kubiatowicz, J.: Toward a scalable, silicon-based quantum computing architecture. IEEE J. Sel. Top. Quantum Electron. 9(6), 1552–1569 (2003)
- Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. Ser. A A439, 553–558 (1992)
- DiVincenzo, D.P.: The physical implementation of quantum computation. Fortsschritte der Physik 48, 771–783 (2000)
- Ekert, A., Hayden, P., Inamori, H.: Basic concepts in quantum computation. 1, 1–37. http://arxiv.org/ pdf/quantph/0011013v1.pdf (2000)
- Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of 28th Annual ACM Symposium on the Theory of Computation, New York, pp. 212–219 (1996)
- Moraga, C.: Using negated control signals in quantum computing circuits. Facta Universitatis (Niš). Ser: Elec. Energ. 24(3), 423–435 (2011)
- Ntalaperas, D., Theodoropoulos, K., Tsakalidis, A., Konofaos, N.: A four base computational method for the implementation of a quantum computer using silicon devices: circuit and simulation. Math. Comput. Model. 51, 144–149 (2010)
- Shepherd, D.J.: On the role of hadamard gates in quantum circuits. Quantum Inf. Process. 5(3), 161– 177 (2006)
- 9. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of 35th Annual Symposium on Foundations of Computer Science, pp. 124–134 (1994)
- Tucker, J.R., Shen, T.-C.: The road to a silicon quantum computer. Quantum Inf. Process. 3(1–5), 105– 113 (2004)