Quantrix: Toward Automated Synthesis of Quantum Cascades

Andrzej Buller

ATR Human Information Science Laboratories
2-2-2 Hikaridai, Seika-Cho, Soraku-gun
Kyoto 619-0288 Japan
buller@atr.co.jp
http://www.atr.co.jp/his/~buller

Abstract

This report provides an idea of a tool for computer aided designing of quantum cascades, preceded by a step-by-step introduction to quantum computing addressed to interdisciplinary students and researchers. Quantum computers, when one day appear, will be able to teleportate information, break secret codes, generate true random numbers, and warn when a message is eavesdropped. Also artificial brain builders must not remain blind to the development of the field of quantum computation. Having to put all necessary computational stuff into robot's head we will supposedly have to employ as many primitive operations as really necessary with possibly low energy dissipation. Reversible circuits dissipate much less energy than the classic ones, while every quantum cascade is reversible. The world of quantum phenomena is also explored in hope to solve the mystery of conscious mind and free will. In order to make readers easily acquire the essence of quantum computation, the presentation is free of distracting quantum-mechanical nomenclature, while any time a new concept is introduced, the full calculation way is provided. Final remarks include a tip how to use the NeuroMaze paradigm to build models of quantum cascades to be run on the ATR's CAM-Brain Machine (CBM).

Contents:

1. Introduction

This report provides brief idea of the Quantrix—a tool for computer aided designing of quantum cascades—preceded by a step-by-step introduction to quantum computing addressed to interdisciplinary students and researchers. Quantum computation is discussed here without reference to quantum mechanics, so the way of presentation does not require prior knowledge of advanced mathematics. It can be said, that this report is a kind of guidelines for programmers who would like to develop the Quantrix. This topic is explored in the framework of the Quantrix Project, launched as one of four themes constituting the Artificial Brain Project conducted at the ATR Human Information Science Laboratories, Kyoto (Buller & Shimohara 2003) in cooperation with the Portland Quantum Logic Group.

Quantum computing, as Williams and Clearwater (1998:xii) noted is currently one of the hottest topics in computer science, physics and engineering. The authors wrote: *Quantum phenomena have no classical analogues. They can be potentially employed to do certain computational tasks much more efficiently tan can be done by any classical computer. Hence, a quantum computer, when one day built, will be able to perform such tasks as teleporting information, breaking supposedly unbreakable codes, generating true random numbers, and communicating with messages that betray the presence of eavesdropping.*

Artificial brain builders must not remain blind to the development of the field of quantum computation. The first reason is that, one day, we will want to put all necessary computational stuff into the head of an intelligent robot. So, we will supposedly have to employ as many primitive operations as really necessary. This idea contradicts the currently dominating computational paradigm based on a processor, RAM, operation system and libraries of standard software. Moreover, the computational technology that got matured by the end of 20th century produces heat, which blocks the way toward 3-dimensional chips. Quantum computing is reversible, which implies a dramatic reduction of energy dissipation (cf. Bennett 1973). The second reason is the quest for machine consciousness. Although the mainstream of politically correct science insist that either the consciousness does not exist at all or it is nothing but a processing of traditional data, there are scientists who explore the world of quantum phenomena in search for an explanation of the mystery of conscious mind and free will (Penrose 1991; Stapp 1993; Ecless 1994).

In order to not to remain unprepared for possible appearance of reversible/quantum hardware (to be used in a way completely different that our good old microprocessors or even FPGAs), artificial brain builders should observe the progress in quantum research and sometimes try to describe mental mechanisms in terms of quantum cascades. No degree in physics is necessary to do this. In this case there is no point to try to contribute to construction of first useful quantum chip. The point is to get understand the difference between classic computing and quantum computing. When the Quantrix appears, people will be able to gain appropriate knowledge just playing with it. This report shows that the math necessary to understand the essentials of quantum computation is much below the academic level. For readers' convenience, any time a new concept is introduced, the full way of calculation is provided, which additionally reminds reader the meaning of related symbols.

Important note: the value 0.71 used in some formulas represents 1 divided by the square root of 2.

---

2. Qubit

Although quantum computation may deal with multiple-state units of information, this report discusses only operations on two-level units called qubits. A qubit (quantum bit) is an entity whose state is defined by an ordered **pair of complex numbers** that meet certain constraint provided below together with an explanation what does it mean 'complex number'.

Let such a pair be denoted as a one-column two-element matrix $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$.

Any matrix $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ is a qubit if and only if

(i) $\alpha_0 = p_0 + q_0 i$, $\alpha_1 = p_1 + q_1 i$, where p_0, q_0, p_0, q_0 are real numbers, while i is so called *imaginary unit* such that $i^2 = -1$,

(ii) $p_0^2 + q_0^2 + p_1^2 + q_1^2 = 1$.

As it can be seen, both α_0 and α_1 contain a "real" element (represented by a real number) and an "imaginary" element (represented by a product of a real number and the imaginary unit i). A complex number is just a number having both real and imaginary element.

When $p_0 = 1$, $q_0 = 0$, $p_1 = 0$, $q_1 = 0$, the state of the qubit is an equivalent of Boolean "0" and can be denoted $|0\rangle$. When $p_0 = 0$, $q_0 = 0$, $p_1 = 1$, $q_1 = 0$, the state of the qubit is an equivalent of Boolean "1" and can be denoted $|1\rangle$. Qubit states $|0\rangle$ and $|1\rangle$ are called *pure states*. When a qubit state has given arbitrary values of p_0, q_0, p_1, q_1, we can consider it as a *superposition* of both pure states weighted by the complex values $p_0 + q_0i$, $p_1 + q_1i$. The complex values are called here *amplitudes*.

3. Unary quantum gates

Every one-input-one-output quantum gate, called unary gate, is a device that changes the state of a given qubit. The new state comes form multiplying a 2×2 transition matrix by the matrix defining the old state. The elements of the transition matrix are complex numbers. In other words:

If $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ and $\begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}$ are the old and new value of a given qubit, respectively, and $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

defines a gate converting $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ into $\begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix}$, then $\begin{bmatrix} \omega_0 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} a\alpha_0 + b\alpha_1 \\ c\alpha_0 + d\alpha_1 \end{bmatrix}$.

If the qubit is processed by a gate defined as $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and, then (immediately) by the

gate defined as $\begin{bmatrix} e & f \\ g & h \end{bmatrix}$ then the resulting state will be $\begin{bmatrix} e & f \\ g & h \end{bmatrix} \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right) =$

$$ = \left(\begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} ea + fc & eb + fd \\ ga + hc & gb + hd \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} $$

Note #3.1: The transition matrix of a concatenation of two quantum gates is a product of the matrix defining the second gate and the matrix defining the first gate. Multiplying the first matrix by the second one would give wrong result.

3.1. Unary *I*-gate

The unary *I*-gate, called identity gate, does not change a qubit state. In schemes it is represented as a horizontal "wire". The only reason of introducing it is that its transition matrix $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is useful for calculating matrices defining more complex structures.

3.2. NOT gate

The NOT gate is represented in schemes as $\oplus$ threaded onto a horizontal wire (Figure 3.1).

The transition matrix of the NOT gate is $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

Hence, the NOT gate converts $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ into $\begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}$.

This means that for pure states of a qubit, the quantum NOT gate behaves as the classic Boolean function NOT (Figure 3.1).

a.
$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0\cdot1 + 1\cdot0 \\ 1\cdot1 + 0\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

b.
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0\cdot0 + 1\cdot1 \\ 1\cdot0 + 0\cdot1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

c.
$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 0\cdot\alpha_0 + 1\cdot\alpha_1 \\ 1\cdot\alpha_0 + 0\cdot\alpha_1 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}$$

Figure 3.1. NOT gate. (a) Conversion of the qubit $|0\rangle$, (b) Conversion of the qubit $|1\rangle$, (c) Conversion of an arbitrary qubit.

3.3. Hadamard gate

The Hadamard gate is represented in schemata as a square with the letter 'H' inside. The square is threaded onto a horizontal wire (Figure 3.2).

The transition matrix of the Hadamard gate is $\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix}$. Hence, the Hadamard

gate converts $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ into $\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix}\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 0.71\alpha_0 + 0.71\alpha_1 \\ 0.71\alpha_0 + (-0.71)\alpha_1 \end{bmatrix} = 0.71\begin{bmatrix} \alpha_0 + \alpha_1 \\ \alpha_0 - \alpha_1 \end{bmatrix}$.

When a qubit $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$ is processed consecutively by two Hadamard gates, the resulting

value will be $\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} \begin{bmatrix} 0.71(\alpha_0 + \alpha_1) \\ 0.71(\alpha_0 - \alpha_1) \end{bmatrix} =$

$$= \begin{bmatrix} 0.71{\cdot}0.71(\alpha_0 + \alpha_1) + 0.71{\cdot}0.71(\alpha_0 - \alpha_1) \\ 0.71{\cdot}0.71(\alpha_0 + \alpha_1) - 0.71{\cdot}0.71(\alpha_0 - \alpha_1) \end{bmatrix} = \begin{bmatrix} 0.5(\alpha_0 + \alpha_1 + \alpha_0 - \alpha_1) \\ 0.5(\alpha_0 + \alpha_1 - \alpha_0 + \alpha_1) \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$

Figure 3.2. Concatenation of two Hadamard gates. (a) Conversion of the qubit $|0\rangle$, (b) Conversion of the qubit $|1\rangle$, (c) Conversion of an arbitrary qubit.

3.4. Square-Root-of-NOT gate

The Square-Root-of-NOT gate is represented in schemata as a square with the letter 'V' inside. The square is threaded onto a horizontal wire (Figure 3.3).

The transition matrix of the Square-Root-of-NOT gate is $\begin{bmatrix} 0.5 + 0.5i & 0.5 - 0.5i \\ 0.5 - 0.5i & 0.5 + 0.5i \end{bmatrix}$.

Hence, the Square-Root-of-NOT gate converts $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ into

$$\begin{bmatrix} 0.5 + 0.5i & 0.5 - 0.5i \\ 0.5 - 0.5i & 0.5 + 0.5i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 + 0.5i \\ 0.5 - 0.5i \end{bmatrix}$$

and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ into $\begin{bmatrix} 0.5 + 0.5i & 0.5 - 0.5i \\ 0.5 - 0.5i & 0.5 + 0.5i \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 - 0.5i \\ 0.5 + 0.5i \end{bmatrix}$

Concatenation of two Square-Root-of-NOT gates is an equivalent of the NOT gate (Figure 3.3).

a.

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \boxed{V} \quad \begin{bmatrix} 0.5 + 0.5i \\ 0.5 - 0.5i \end{bmatrix} \quad \boxed{V} \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

b.

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \boxed{V} \quad \begin{bmatrix} 0.5 - 0.5i \\ 0.5 + 0.5i \end{bmatrix} \quad \boxed{V} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

c.

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \quad \boxed{V} \quad 0.5 \begin{bmatrix} \alpha_0 + \alpha_1 + (\alpha_0 - \alpha_1)\,i \\ \alpha_0 + \alpha_1 - (\alpha_0 - \alpha_1)\,i \end{bmatrix} \quad \boxed{V} \quad \begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}$$
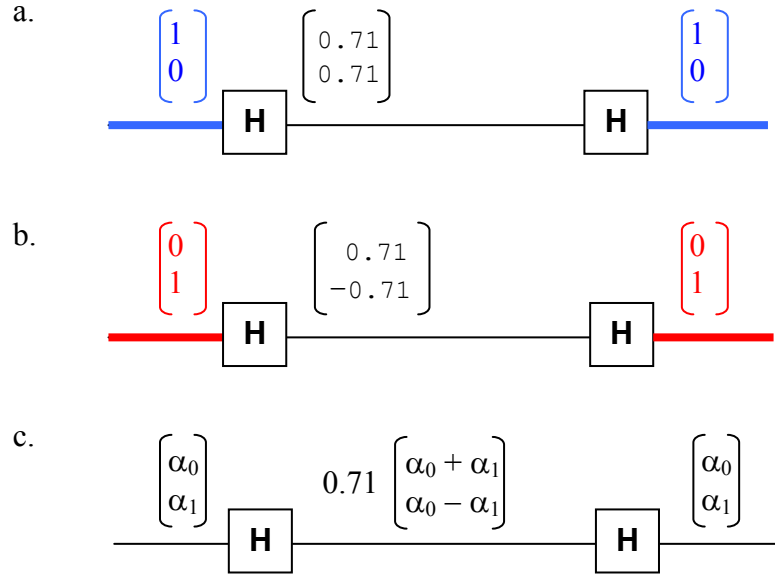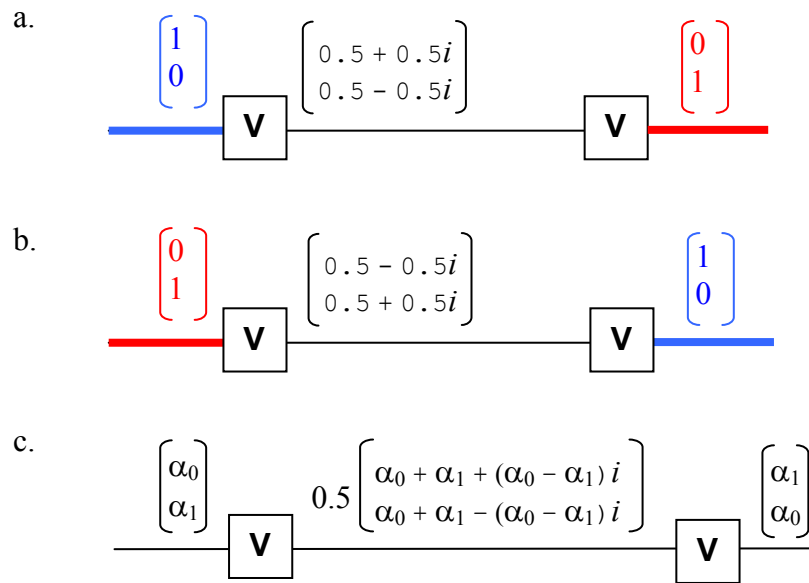
Figure 3.3. Concatenation of two Square-Root-of-NOT gates.
(a) Conversion of the qubit $|0\rangle$, (b) Conversion of the qubit $|1\rangle$,
(c) Conversion of an arbitrary qubit.

In order to make sure that the concatenation of two V-gates behaves as the quantum NOT gate, let us use recall the Note #3.1 and calculate:

$$\begin{bmatrix} 0.5 + 0.5i & 0.5 - 0.5i \\ 0.5 - 0.5i & 0.5 + 0.5i \end{bmatrix} \begin{bmatrix} 0.5 + 0.5i & 0.5 - 0.5i \\ 0.5 - 0.5i & 0.5 + 0.5i \end{bmatrix} =$$

$$= 0.5 \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \; 0.5 \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} =$$

$$= 0.25 \begin{bmatrix} (1+i)(1+i)+(1-i)(1-i) & (1+i)(1-i)+(1-i)(1+i) \\ (1-i)(1+i)+(1+i)(1-i) & (1-i)(1-i)+(1+i)(1+i) \end{bmatrix} =$$

$$= 0.25 \begin{bmatrix} (1+2i+i^2)+(1-2i+i^2) & 2(1-i^2) \\ 2(1-i^2) & (1+2i+i^2)+(1-2i+i^2) \end{bmatrix} =$$

$$= 0.25 \begin{bmatrix} 1+i^2+1+i^2 & 2-2i^2 \\ 2-2i^2 & 1+i^2+1+i^2 \end{bmatrix} = 0.5 \begin{bmatrix} 1+i^2 & 1-i^2 \\ 1-i^2 & 1+i^2 \end{bmatrix} =$$

$$= 0.5 \begin{bmatrix} 1+i^2 & 1-i^2 \\ 1-i^2 & 1+i^2 \end{bmatrix} = 0.5 \begin{bmatrix} 1+(-1) & 1-(-1) \\ 1-(-1) & 1+(-1) \end{bmatrix} = 0.5 \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

i.e. just as for the NOT gate! ⟶

7

3.5. V† gate

The "mirror" of Square-Root-of-NOT gate is represented in schemata as a square with the symbol 'V†' inside. The square is threaded onto a horizontal wire (Figure 3.4).

The transition matrix of the V† gate is $\begin{bmatrix} 0.5 - 0.5i & 0.5 + 0.5i \\ 0.5 + 0.5i & 0.5 - 0.5i \end{bmatrix}$.

Hence, V† gate converts $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ into $\begin{bmatrix} 0.5 - 0.5i \\ 0.5 + 0.5i \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ into $\begin{bmatrix} 0.5 + 0.5i \\ 0.5 - 0.5i \end{bmatrix}$.

Concatenation of two V† gates is an equivalent of the NOT gate, while concatenation of one V gate and one V† gate returns the initial qubit state (Figure 3.4).
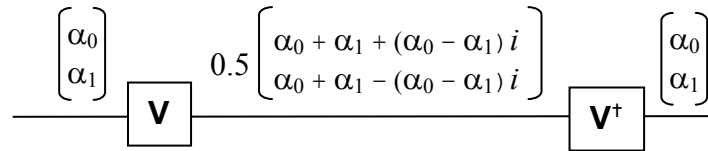
Figure 3.4. Concatenation of one V gate and one V† gate returns the initial qubit state.

3.6. Pauli gates

The set of Pauli gates contains four gates defined by 2×2 matrices I, σ_X, σ_Y, and σ_Z.

$$\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

As it can be noted, σ_X is identical with the NOT gate, while I is the unary identity gate introduced before.

4. Two-qubit register

A pair of qubits constitutes a *2-qubit register* represented as an ordered quadruple of complex numbers such that the sum of the squares of the modules of the numbers is equal to 1. We well denote such a quadruple as a one-column four-element matrix calculated as Kronecker product of related qubits (Figure 4.1 and 4.2).

8

$$
\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \cdots\cdots\rightarrow \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \otimes \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \\ \alpha_1 \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix}
$$

$$
\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \cdots\cdots\cdots\cdots\cdots\cdots\cdots\uparrow
$$

Figure 4.1. Two qubits make a 2-qubit register ($\otimes$ - Kronecker product)

$$
\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\qquad
\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}
$$

$$
\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}
\qquad
\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

Figure 4.2. States of a 2-qubit register for pure states of contributing qubits.

A 2-qubit register, as well as any n-qubit register, can get into a state that is not decomposable into states of contributing qubits. In such a case we say that the state is an *entangled state*. An example of an entangled state is shown in Figure 4.3.

Task: Decompose the state of a 2-qubit register
$$
\begin{bmatrix} 0.71 \\ 0.00 \\ 0.00 \\ 0.71 \end{bmatrix}
$$

i.e. find such $\alpha_0, \alpha_1, \beta_0, \beta_1$ that
$$
\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \otimes \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 0.71 \\ 0.00 \\ 0.00 \\ 0.71 \end{bmatrix}
$$

In other words find such $\alpha_0, \alpha_1, \beta_0, \beta_1$ that
$$
\begin{cases} \alpha_0\beta_0 = 0.71 \\ \alpha_0\beta_1 = 0.00 \\ \alpha_1\beta_0 = 0.00 \\ \alpha_1\beta_1 = 0.71 \end{cases}
$$

Unfortunately, such $\alpha_0, \alpha_1, \beta_0, \beta_1$ do not exist, which means that the state we attempted to decompose is entangled.

Figure 4.3. An example of an entangled state of a 2-qubit register

5. Two-qubit gates

Every two-input-two-output quantum gate (called by some authors 'binary gate' which seems to be a bit confusing) is a device that changes the state of a 2-qubit register. The new register state comes form multiplying a 4×4 transition matrix by the matrix defining the old state of the register. The elements of the transition matrix are complex numbers. In other words:

If $\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$ and $\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}$ are the old and new state of a 2-qubit register, respectively,

and $\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$ defines a gate converting $\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$ into $\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix}$,

then $\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} a\mu_0 + b\mu_1 + c\mu_2 + d\mu_3 \\ e\mu_0 + f\mu_1 + g\mu_2 + h\mu_3 \\ i\mu_0 + j\mu_1 + k\mu_2 + l\mu_3 \\ m\mu_0 + n\mu_1 + o\mu_2 + p\mu_3 \end{pmatrix}.$

5.1. Two-qubit *I*-gate

The two-qubit *I*-gate, called identity gate, does not change a 2-qubit register state.

The only reason of introducing it is that its transition matrix $I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

is useful for calculating matrices defining more complex structures.

5.2. CNOT (Feynman) Gate

The CNOT (Controlled NOT) gate, called also the Feynman gate, applies to 2-qubit register, so in drawings it concerns two and only two wires. It is represented using a compound of three symbols: $\oplus$, $\bullet$, and | that represent an inverter, a control and a connection, respectively (Figure 5.1). The qubit that is associated with the control is called *control qubit*. The qubit that is associated with the inverter is called *target qubit*.

The transition matrix of the quantum CNOT gate is $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$.

Hence, the quantum CNOT gate converts $\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$ into

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} 1 \cdot \mu_0 + 0 \cdot \mu_1 + 0 \cdot \mu_2 + 0 \cdot \mu_3 \\ 0 \cdot \mu_0 + 1 \cdot \mu_1 + 0 \cdot \mu_2 + 0 \cdot \mu_3 \\ 0 \cdot \mu_0 + 0 \cdot \mu_1 + 0 \cdot \mu_2 + 1 \cdot \mu_3 \\ 0 \cdot \mu_0 + 0 \cdot \mu_1 + 1 \cdot \mu_2 + 0 \cdot \mu_3 \end{pmatrix} = \begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_3 \\ \mu_2 \end{pmatrix}$$
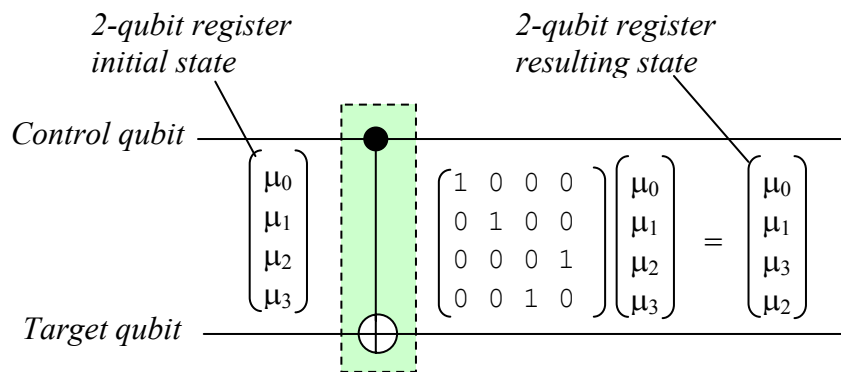
Figure 5.1. Quantum CNOT (Feynman) gate

When the control qubit in the pure state $|0\rangle$ the target qubit remains unchanged (Figure 5.2a). When the control qubit in the pure state $|1\rangle$, the target qubit is processed the same way as using the quantum NOT gate (Figure 5.2b).

For both qubits in their pure states the quantum CNOT works the same way as the classic CNOT (Figure 5.3). It can be noted that if the target qubit is $|0\rangle$, the CNOT gate behaves as a fan-out element (cf. Figure 5.3, *Case 1* and *Case 2*). Unfortunately, this "fan-out" will not work for arbitrary state of the target qubit (see Figures 5.4 and 5.5).

5.2. SWAP Gate

The SWAP gate applies to 2-qubit register, so in drawings it concerns two and only two wires. It is represented using two copies of the symbol ×, and one copy of | that mark the states to be swapped and a connection, respectively (Figure 5.6).

a.

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

b.

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ \beta_0 \\ \beta_1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \beta_1 \\ \beta_0 \end{bmatrix}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix}$$
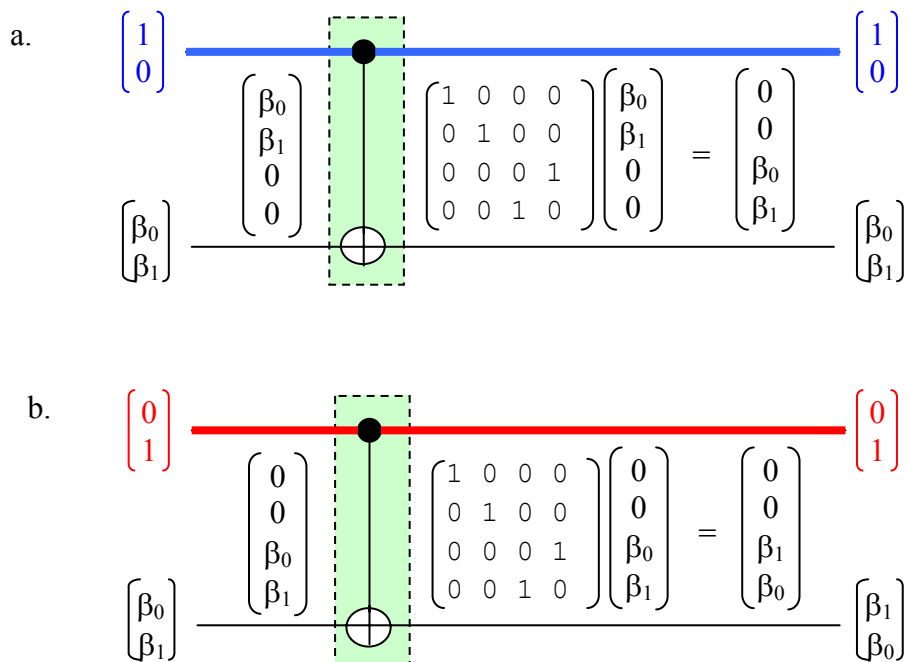
Figure 5.2. Quantum CNOT's behavior for pure control states

Case 1

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Case 2

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Case 3

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Case 4

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
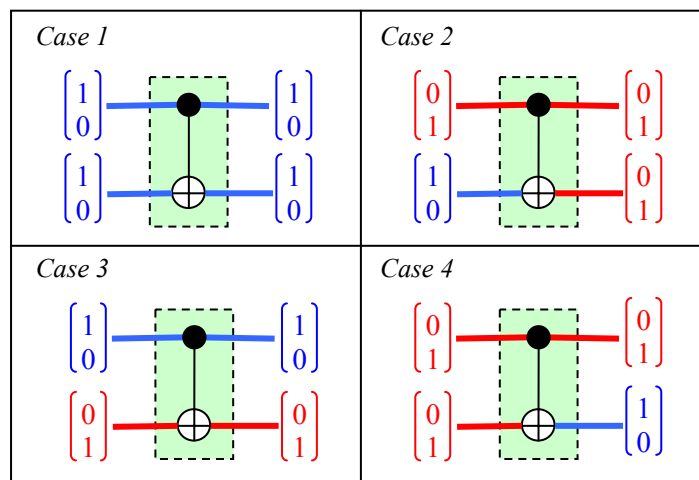
Figure 5.3. Quantum CNOT's behavior for all pure states

$|y\rangle$ ———— $|y\rangle$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$ ———— $|y\rangle$

Figure 5.4. CNOT (Feynman) gate as a quantum "fan-out". Unfortunately, such a "fan-out" works only if $|y\rangle$ is a pure state of the control qubit.

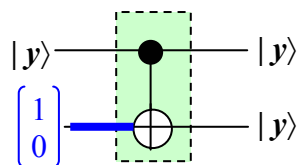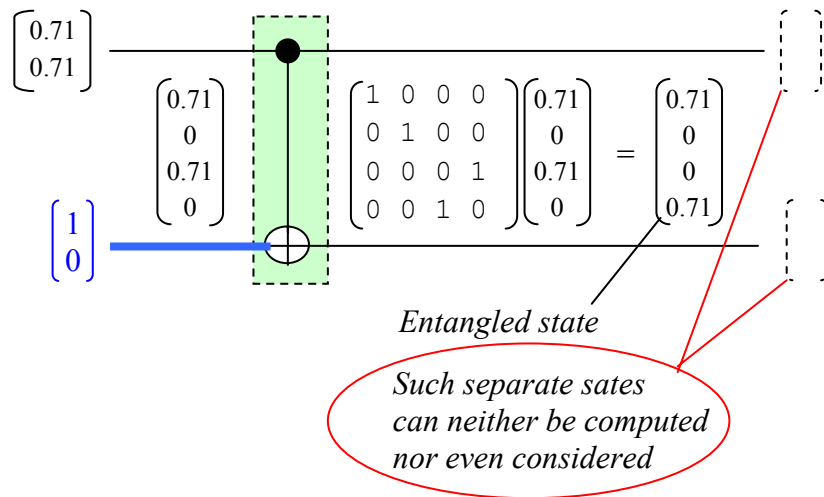Figure 5.5. An attempt to fan-out a non-pure state of the control qubit resulted in an entangled state of the entire 2-qubit register. Indeed, when the register is in an entangled state, the states of contributing qubits must not be treated separately. One can make use of this astonishing property but this is beyond the scope of this report.

The SWAP gate converts $\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \otimes \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$ into $\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \otimes \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$. This means that the initial

and resulting state of a related 2-qubit register are $\begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix} , \begin{pmatrix} \beta_0\alpha_0 \\ \beta_0\alpha_1 \\ \beta_1\alpha_0 \\ \beta_1\alpha_1 \end{pmatrix}$, respectively.

Such operation can be performed by matrix $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.
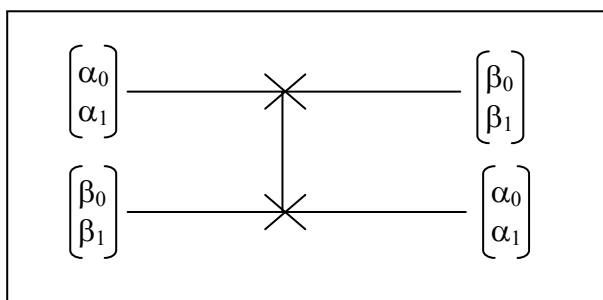

Figure 5.6. SWAP gate

13

5.3. Controlled-V gate

The Controlled-V (CV) gate, applies to 2-qubit register, so in drawings it concerns two and only two wires. It is represented using a compound of three symbols: a square with the letter 'V' inside, •, and | that represent the Square-Root-of-NOT, a control and a connection, respectively (Figure 5.7a). The qubit that is associated with the control is called *control qubit*. The qubit that is associated with the Square-Root-of-NOT is called *target qubit*.

The transition matrix of the CV gate is $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5+0.5i & 0.5-0.5i \\ 0 & 0 & 0.5-0.5i & 0.5+0.5i \end{pmatrix}$.

5.4. Controlled-V† gate

The Controlled-V (CV†) gate, applies to 2-qubit register, so in drawings it concerns two and only two wires. It is represented using a compound of three symbols: a square with the symbol 'V†' inside, •, and | that represent the "mirror" of Square-Root-of-NOT, a control and a connection, respectively (Figure 5.7b). The qubit that is associated with the control is called *control qubit*. The qubit that is associated with the Square-Root-of-NOT is called *target qubit*.

The transition matrix of the CV† gate is $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5-0.5i & 0.5+0.5i \\ 0 & 0 & 0.5+0.5i & 0.5-0.5i \end{pmatrix}$.
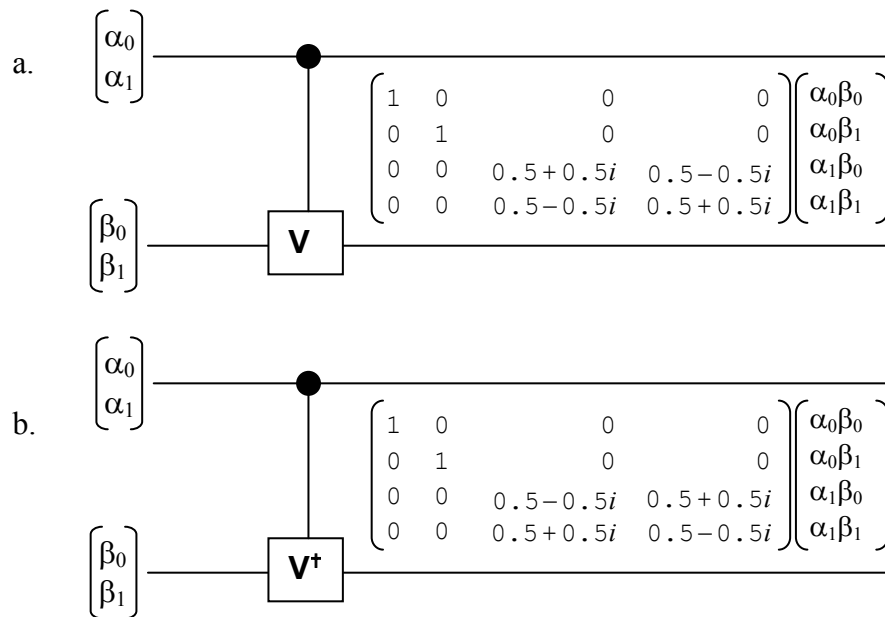

Figure 5.7. Controlled Square-Root-of-NOT gates

5.5. Concatenations of binary quantum gates

Let us consider the concatenation of two gates shown in Figure 5.8.

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \quad \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix}$$
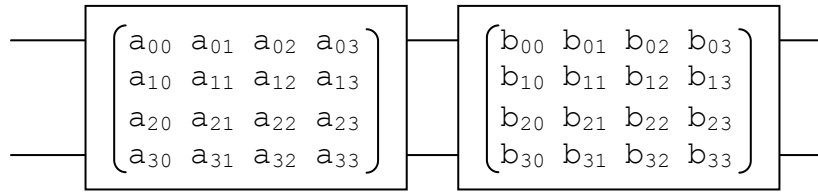
Figure 5.8. Concatenation of two binary quantum gates
defined by 4×4 transition matrices.

It processes the state of a 2-qubit register as if it were a single gate defined using the
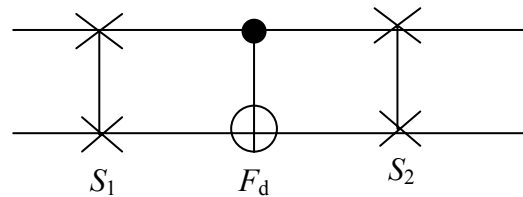
matrix:
$$\begin{pmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}$$

where $c_{ij} = b_{i0}a_{0j} + b_{i1}a_{1j} + b_{i2}a_{2j} + b_{i3}a_{3j}$.

Let us consider the scheme shown in Figure 5.9.

Figure 5.9. Concatenation
of three binary quantum gates

$S_1 \qquad F_d \qquad S_2$

The concatenation of three binary quantum gates will behave as a gate defined by the
matrix $F_u = S_2 (F_d S_1) =$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

What kind of register-state change is defined by matrix F_u? We could calculate four
cases of register pure states, but let as use a trick. Let us draw the SWAP gates as they
were true devices made of true wires (Figure 5.10a). When we now stretch the wires to

15

make them straight, the Feynman gate will flip vertically (5.10b). This way we obtained the transition matrix for flipped Feynman gate.
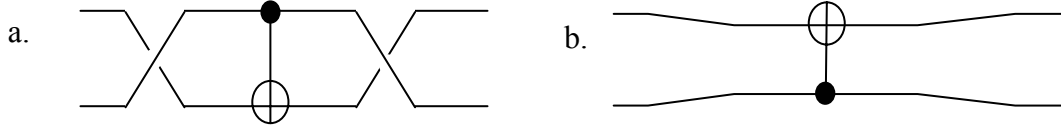

a.   b.

Figure 5.10. Flipped Feynman gate as a concatenation of "non-flipped" Feynman gate with two SWAP gates (the "wires" are only a metaphor).

5.6. Joint unary gates

A two-input two-output gate can be composed of two parallel unary gates.

If the upper unary gate is defined by the matrix $G_0 = \begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix}$

and the lower unary gate is defined by the matrix $G_1 = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}$

then the transition matrix of the pair of the gates is the *Kronecker product* of G_0 and G_1, i.e.

$$G = \begin{bmatrix} a_0 & b_0 \\ c_0 & d_0 \end{bmatrix} \otimes \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} = \begin{pmatrix} a_0 \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} & b_0 \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \\ c_0 \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} & d_0 \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \end{pmatrix} = \begin{pmatrix} a_0 a_1 & a_0 b_1 & b_0 a_1 & b_0 b_1 \\ a_0 c_1 & a_0 d_1 & b_0 c_1 & b_0 d_1 \\ c_0 a_1 & c_0 b_1 & d_0 a_1 & d_0 b_1 \\ c_0 c_1 & c_0 d_1 & d_0 c_1 & d_0 d_1 \end{pmatrix}$$

As the first example let us calculate the transition matrix of a "wire" put in parallel with the NOT gate (Figure 5.11).

Since the matrix for "wire" is $I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$,

While the matrix for the NOT gate is $N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

then the joint transition matrix of the pair "wire" || NOT of the gates is the *Kronecker product* of I_2 and N, i.e.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{pmatrix} 1 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 0 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ 0 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 1 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$
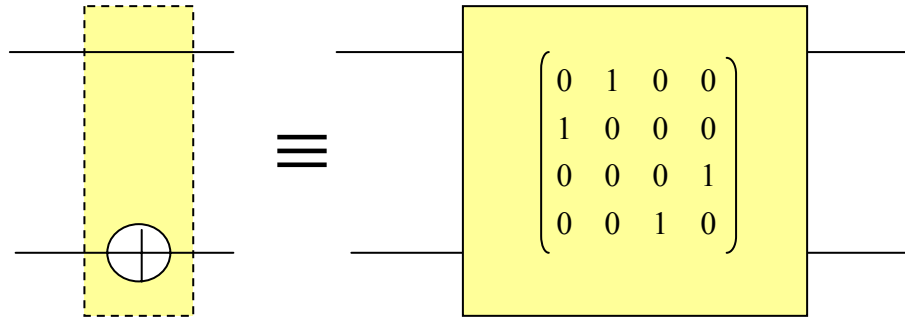
Figure 5.11. NOT gate coupled in parallel with a „wire”.

As another example let us take two parallel Hadamard gates (Figure 5.12). Their joint behavior can be described in terms of a single two-input two-output gate with the transition matrix calculated as follows:

$$\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} \otimes \begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} = \begin{bmatrix} 0.71\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} & 0.71\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} \\ 0.71\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} & -0.71\begin{bmatrix} 0.71 & 0.71 \\ 0.71 & -0.71 \end{bmatrix} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Figure 5.12. An example of behavior of two parallel Hadamard gates.

6. Three-qubit register

A triple of qubits constitutes a *3-qubit register* represented as a one-column eight-element matrix of complex numbers such that the sum of the squares of the modules of the numbers is equal to 1. The matrix calculated as Kronecker product of related qubits (Figure 6.1).
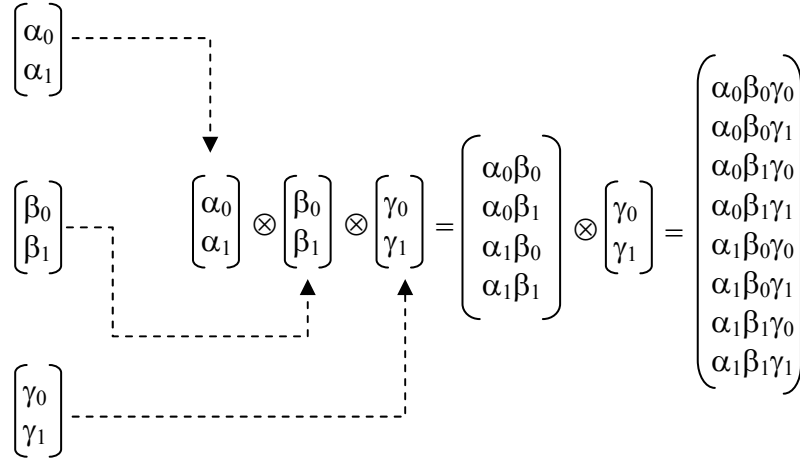
Figure 6.1. Three qubits constituting a 3-qubit register
($\otimes$ - symbol of Kronecker product)

7. Three-input-three-output quantum gates (three-qubit gates)

Every three-input-three-output quantum gate is a device that changes the state of a 3-qubit register. The new register state comes form multiplying an 8×8 transition matrix by the matrix defining the old state of the register. In other words:

If $\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}$ and $\begin{pmatrix} \nu_0 \\ \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \end{pmatrix}$ are the old and new state of a 3-qubit register, respectively, and

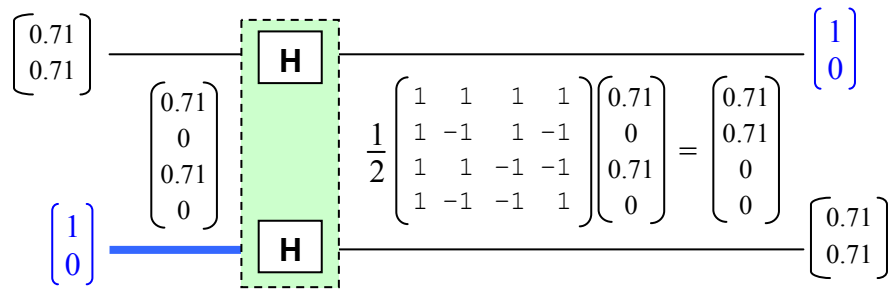$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} & a_{06} & a_{07} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} \\ a_{60} & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} \\ a_{70} & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} \end{pmatrix}$$ defines a gate converting $\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}$ into $\begin{pmatrix} \nu_0 \\ \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \\ \nu_5 \\ \nu_6 \\ \nu_7 \end{pmatrix}$,

then $\begin{pmatrix} \nu_0 \\ \nu_1 \\ \vdots \\ \nu_7 \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & \cdots & a_{07} \\ a_{10} & a_{11} & \cdots & a_{17} \\ \vdots & \vdots & \ddots & \vdots \\ a_{70} & a_{71} & \cdots & a_{77} \end{pmatrix} \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_7 \end{pmatrix} = \begin{pmatrix} a_{00}\mu_0 + a_{01}\mu_1 + \cdots + a_{07}\mu_3 \\ a_{10}\mu_0 + a_{11}\mu_1 + \cdots + a_{17}\mu_3 \\ \vdots \\ a_{70}\mu_0 + a_{71}\mu_1 + \cdots + a_{77}\mu_3 \end{pmatrix}$.

18

7.1. Concatenation of 3-input 3-output quantum gates

Let us consider the concatenation of two gates shown in Figure 7.1.

$$
\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{07} \\ a_{10} & a_{11} & \cdots & a_{17} \\ \vdots & \vdots & \ddots & \vdots \\ a_{70} & a_{71} & \cdots & a_{77} \end{pmatrix}
\qquad
\begin{pmatrix} b_{00} & b_{01} & \cdots & b_{07} \\ b_{10} & b_{11} & \cdots & b_{17} \\ \vdots & \vdots & \ddots & \vdots \\ b_{70} & b_{71} & \cdots & b_{77} \end{pmatrix}
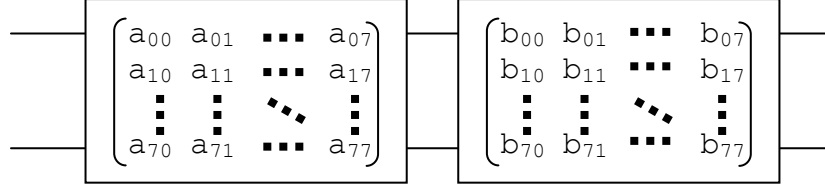$$

Figure 7.1. Concatenation of two 3-input 3-output quantum gates

It processes the state of a 3-qubit register as if it were a single gate defined using the matrix:

$$
\begin{pmatrix} c_{00} & c_{01} & \cdots & c_{07} \\ c_{10} & c_{11} & \cdots & c_{17} \\ \vdots & \vdots & \ddots & \vdots \\ c_{70} & c_{71} & \cdots & c_{77} \end{pmatrix}
=
\begin{pmatrix} b_{00} & b_{01} & \cdots & b_{07} \\ b_{10} & b_{11} & \cdots & b_{17} \\ \vdots & \vdots & \ddots & \vdots \\ b_{70} & b_{71} & \cdots & b_{77} \end{pmatrix}
\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{07} \\ a_{10} & a_{11} & \cdots & a_{17} \\ \vdots & \vdots & \ddots & \vdots \\ a_{70} & a_{71} & \cdots & a_{77} \end{pmatrix}
$$

where $c_{ij} = b_{i0}a_{0j} + b_{i1}a_{1j} + b_{i2}a_{2j} + b_{i3}a_{3j} + \ldots + b_{i7}a_{7j}$.

7.2. WCI (Wire||Control||Inverter) gate

Let us consider the Feynman-based 3-input 3-output gate shown in Figure 7.2. The transition matrix F_{wci} is calculated as Kronecker product of I_2 and F_{ci}. Hence,

$$
F_{WCI} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & 0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ 0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & 1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}
$$

$$
\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
\qquad
\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}
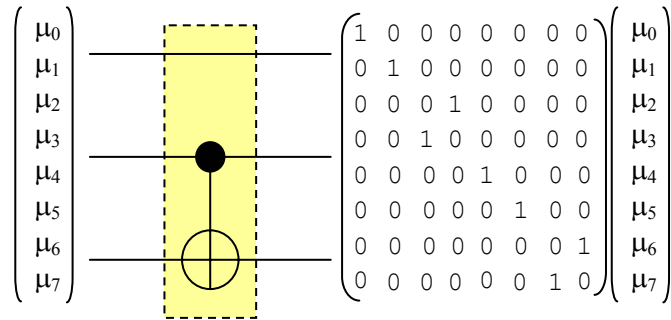\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
$$

Figure 7.2. Feynman-based 3-input 3-output WCI (Wire-Control-Inverter) gate

7.3. CIW (Control‖Inverter‖Wire) gate

Another Feynman-based 3-input 3-output gate is shown in Figure 7.3. The transition matrix F_{CIW} is calculated as Kronecker product of F_{CI} and I_2. Hence,

$$
F_{CIW} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{pmatrix} 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \end{pmatrix}
$$

$$
\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix} \quad \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \end{pmatrix} \begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
$$

Figure 7.3. Feynman-based 3-input 3-output CIW (Control-Inverter-Wire) gate

7.4. XXI (SWAP‖Wire) gate

A SWAP-based 3-input 3-output gate is shown in Figure 7.4. The transition matrix S' is calculated as Kronecker product of S and I_2. Hence,

$$
S' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{pmatrix} 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} \\ 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 0\begin{bmatrix}1&0\\0&1\end{bmatrix} & 1\begin{bmatrix}1&0\\0&1\end{bmatrix} \end{pmatrix}
$$

$$
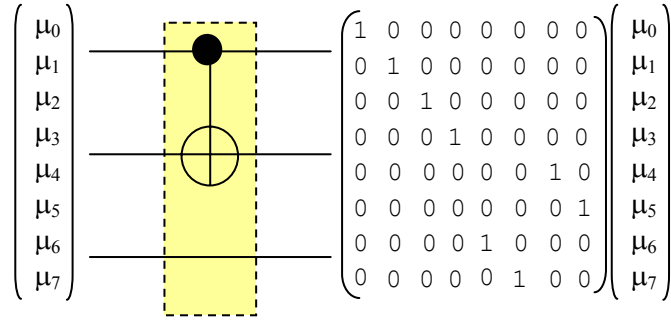\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
\quad
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
$$

Figure 7.4. SWAP-based 3-input 3-output S′ gate

7.5. WCV (Wire‖Control‖Square-Root-of-NOT) gate

Let us consider the Square-Root-of-NOT-based 3-input 3-output gate shown in Figure 7.5. The transition matrix F_{WCV} is calculated as Kronecker product of I_2 and F_{CV}. Hence,

$$
F_{WCI} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & q \\ 0 & 0 & q & p \end{pmatrix}
=
\begin{pmatrix}
1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & q \\ 0 & 0 & q & p \end{bmatrix} & 0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & q \\ 0 & 0 & q & p \end{bmatrix} \\
0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & q \\ 0 & 0 & q & p \end{bmatrix} & 1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & q \\ 0 & 0 & q & p \end{bmatrix}
\end{pmatrix}
=
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & p & q & 0 & 0 & 0 & 0 \\
0 & 0 & q & p & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & p & q \\
0 & 0 & 0 & 0 & 0 & 0 & q & p
\end{pmatrix}
$$

where:
p = 0.5+0.5i
q = 0.5−0.5i

$$
\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
\quad
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & p & q & 0 & 0 & 0 & 0 \\
0 & 0 & q & p & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & p & q \\
0 & 0 & 0 & 0 & 0 & 0 & q & p
\end{pmatrix}
\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}
$$

Figure 7.5. V-gate-based 3-input 3-output WCV (Wire-Control-V) gate
(p = 0.5+0.5i, q = 0.5−0.5i)

7.6. WCV† (Wire‖Control‖Square-Root-of-NOT†) gate

Let us consider the Square-Root-of-NOT†-based 3-input 3-output gate shown in Figure 7.6. The transition matrix $F_{\text{WCV+}}$ is calculated as Kronecker product of I_2 and $F_{\text{CV+}}$. Hence,

$$
F_{\text{WCI}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & p & q \end{bmatrix} = \begin{pmatrix} 1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & p & q \end{bmatrix} & 0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & p & q \end{bmatrix} \\ 0\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & p & q \end{bmatrix} & 1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & q & p \\ 0 & 0 & p & q \end{bmatrix} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q & p & 0 & 0 & 0 & 0 \\ 0 & 0 & p & q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q & p \\ 0 & 0 & 0 & 0 & 0 & 0 & p & q \end{pmatrix}
$$

where:
p = 0.5+0.5*i*
q = 0.5-0.5*i*
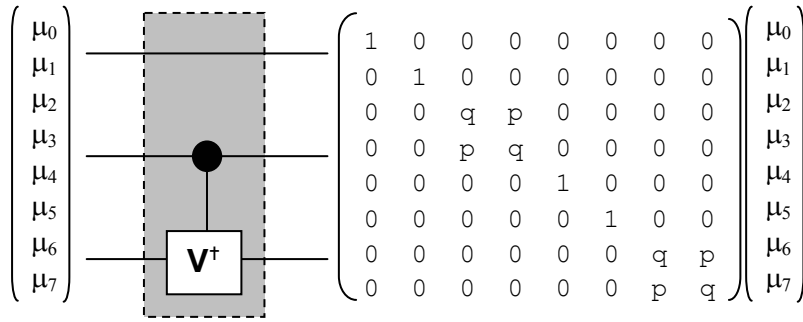
Figure 7.6. V†-gate-based 3-input 3-output WCV† (Wire-Control-V†) gate
$\left(\text{p} = 0.5+0.5i, \text{q} = 0.5-0.5i \right)$
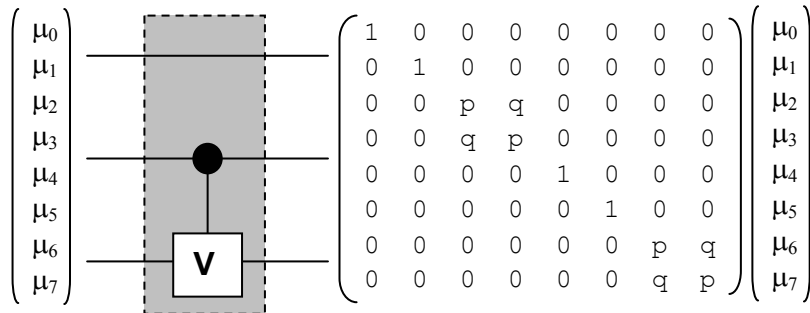
7.7. CWV (Wire‖Control‖Square-Root-of-NOT) gate

Let us consider the Square-Root-of-NOT-based 3-input 3-output gate shown in Figure 7.7. The transition matrix F_{CWV} is calculated as a product $S'F_{\text{CV}}S'$.
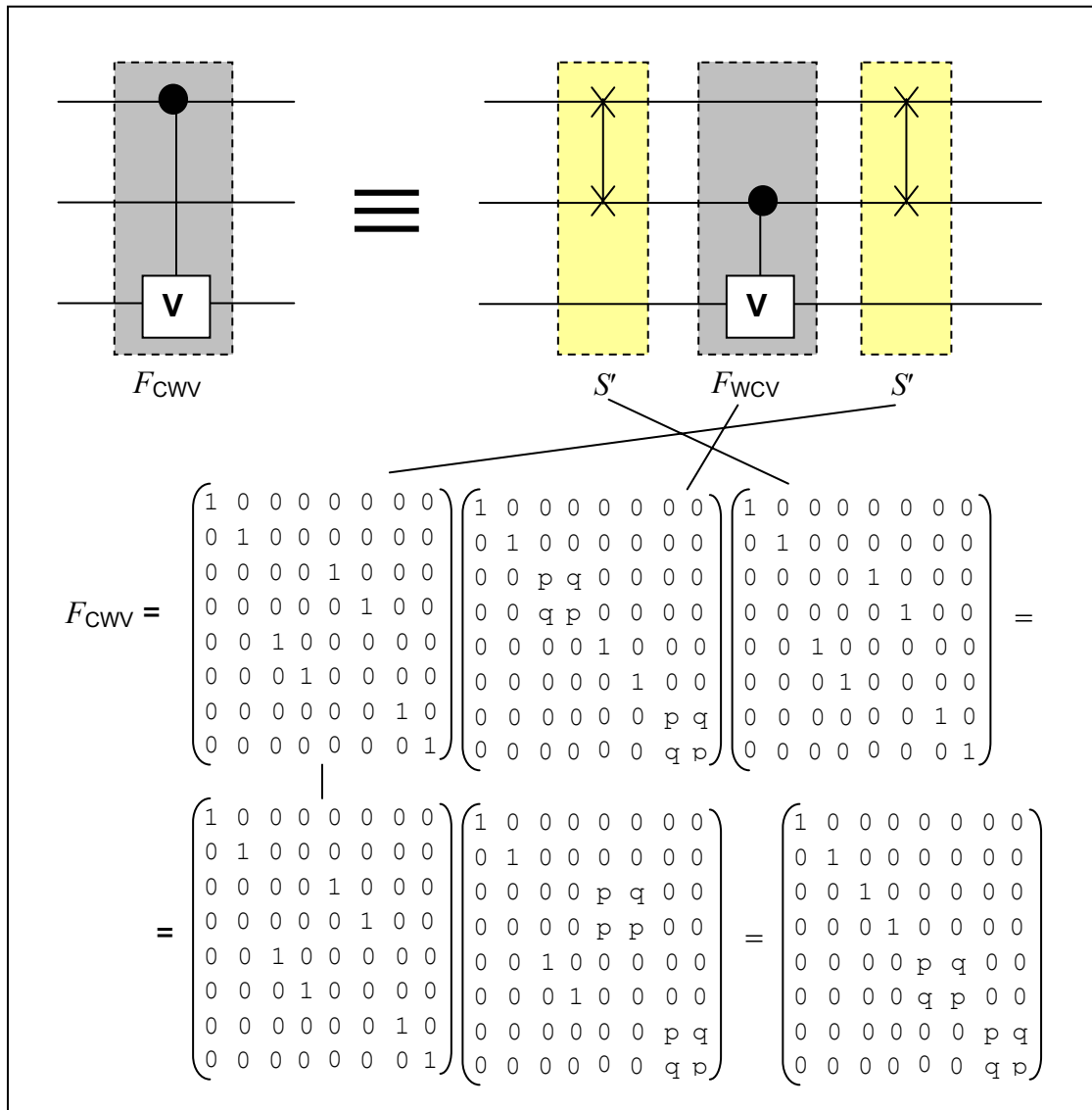
Figure 7.7. V-gate-based 3-input 3-output CWV (Control-Wire -V) gate
$$(p = 0.5+0.5i, q = 0.5-0.5i)$$

$$F_{CWV} = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&p&q&0&0&0&0 \\ 0&0&q&p&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&p&q \\ 0&0&0&0&0&0&q&p \end{pmatrix} \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix} \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&p&q&0&0 \\ 0&0&0&0&p&p&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&p&q \\ 0&0&0&0&0&0&q&p \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&p&q&0&0 \\ 0&0&0&0&q&p&0&0 \\ 0&0&0&0&0&0&p&q \\ 0&0&0&0&0&0&q&p \end{pmatrix}$$

7.8. C^2NOT (Toffoli) Gate

The C^2NOT gate, called also Toffoli gate, applies to 3-qubit register, so in drawings it concerns three and only three wires. It is represented using a compound of four graphic elements: $\oplus$, two copies of $\bullet$, and $|$ that represent an inverter, two controls and a connection, respectively (Figure 3.13). The qubits that is associated with the controls are called *control qubist*. The qubit that is associated with the inverter is called *target qubit*.

23

$$\begin{pmatrix} \mu_0 \\ \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\ \mu_5 \\ \mu_6 \\ \mu_7 \end{pmatrix}$$
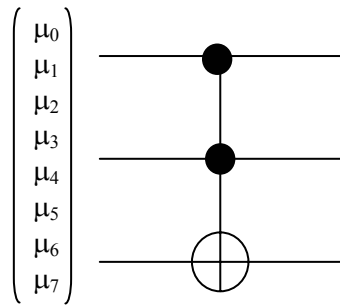
Figure 7.8. Toffoli gate

The transition matrix of the quantum C^2NOT gate is $T_{CCI} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$.

Let us check the Toffoli gate's behavior for four cases with pure initial states:

Ad Case 1

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Ad Case 2

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Ad Case 3

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Ad Case 3

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
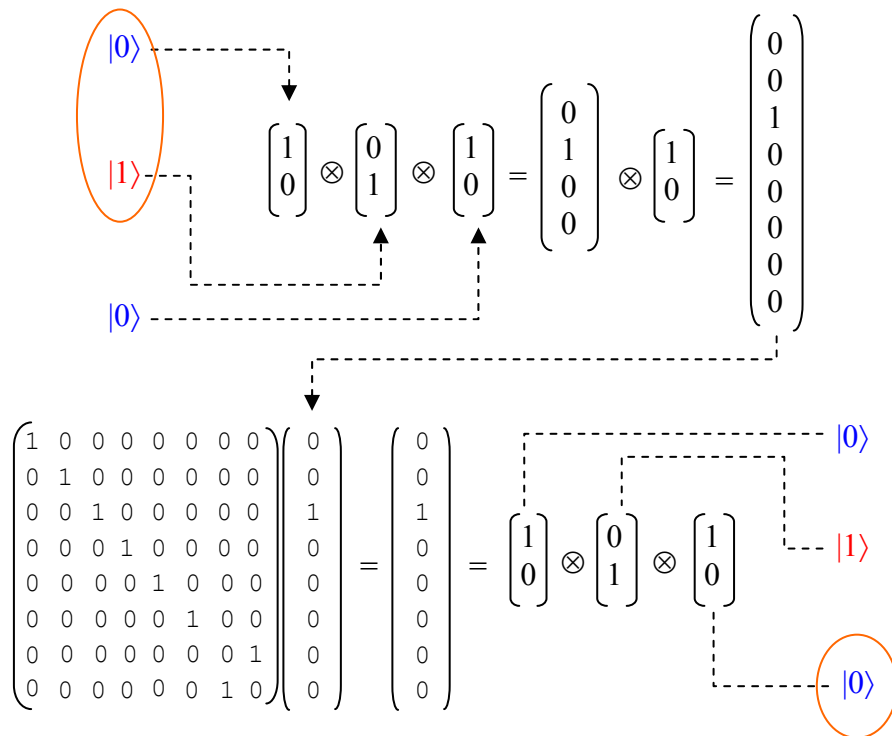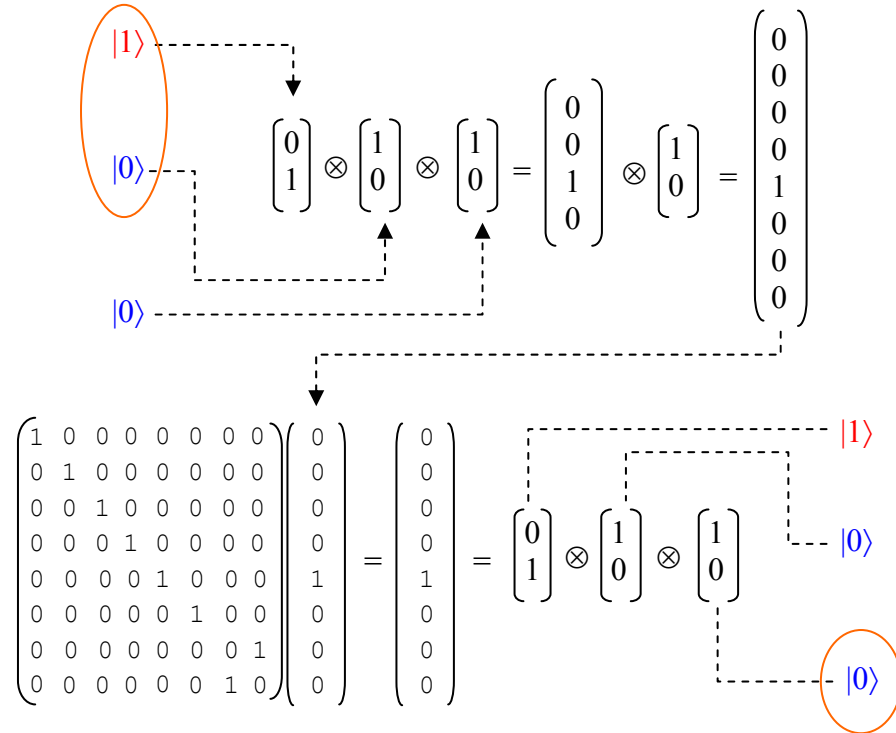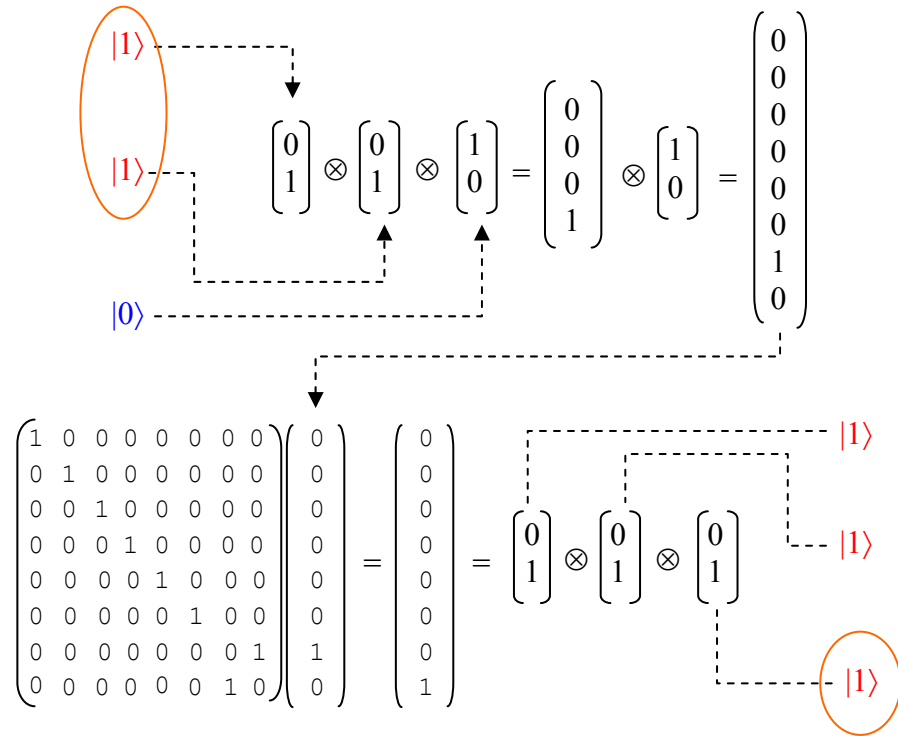
The summary of the checking is shown in Figure 7.9. As it can be seen, for a 3-qubit register in its pure states when the third qubit is in constant state $|0\rangle$, the Tofoli gate may be used as the classic AND gate.
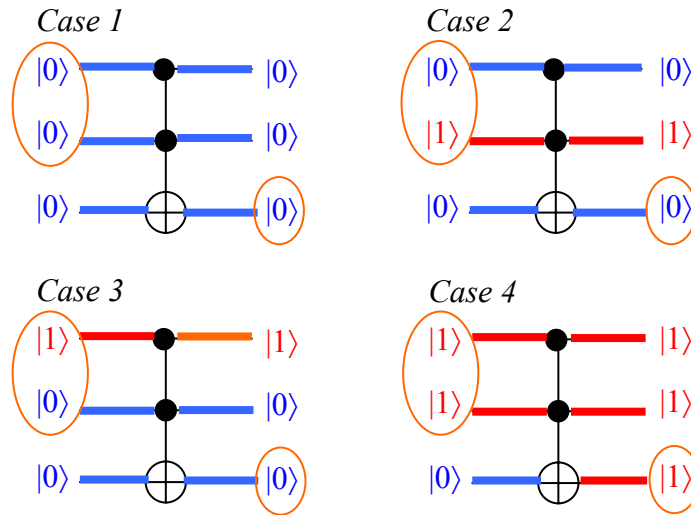

Figure 7.9. For pure states of a 3-qubit register, the Toffoli gate may be used as a classic AND gate.

8. Quantum cascades

Unlike some sorts of binary (2-input 2-output) gates, a physical implementation of any 3-qubit gates as a compact device is an open question and even no convincing ideas has been reported in the matter. Hence, one of directions of quantum computing development is search for methods of automated synthesis of *n*-qubit gates (for *n*>2) based on a limited set of simple 1-qubit or 2-qubut gates. Figure 8.1 shows a cascade being an equivalent of a Toffoli gate.
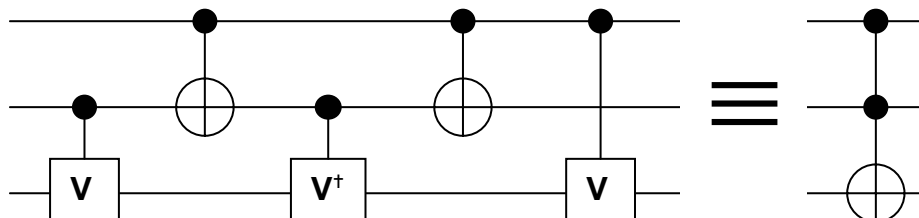

Figure 8.1. Toffoli gate as a cascade of simple quantum gates

The equivalence can be checked via multiplying matrices defining the binary-gate-based 3-input 3-output gates (Figure 8.2).
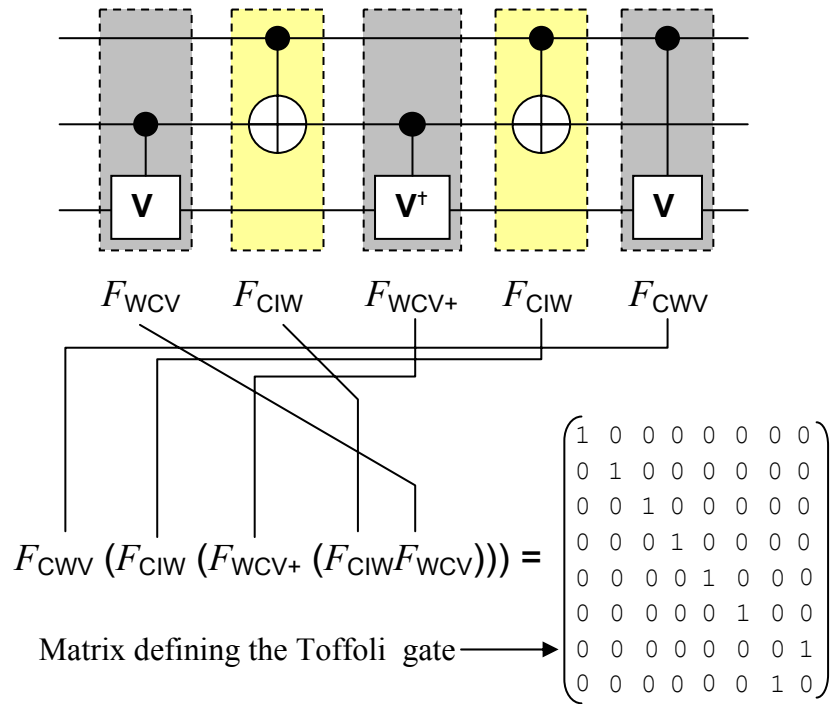

$$F_{CWV} (F_{CIW} (F_{WCV+} (F_{CIW}F_{WCV}))) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Matrix defining the Toffoli gate $\longrightarrow$

Figure 8.2. Proof of correctness of the cascade substituting the Toffoli gate. The matrices F_{WCV}, F_{WCV+}, F_{CIW} and F_{CWV} were introduced in Chapter 7.

9. Quantum cascade synthesis

The Quantrix is a software tool for computer aided design of quantum cascades. In this chapter the first outline of interfacing is provided. The key concepts are the Worksheet and the Matrix. The Worksheet is a grid of N horizontal lines and M vertical lines. The user can drag a desired 1-qubit gate and drop it in any node of the grid (Figure 9.1). A relevant Matrix appears immediately and changes any time a next function is dragged and dropped (Figure 9.3). In order to make the interface user friendly elements of the Matrix are represented as colored squares according to a user defined mapping. A recommended mapping is shown in Figure 9.2. The user can also create a Matrix for unknown cascade and the Quantrix will employ a search method to provide a cascade represented by the Matrix (Figure 9.4). The search is scientific challenge. The Portland Quantum Logic Group reports some promising results in evolving quantum cascades (Lucas & Perkowski 2002; Lukas et al. 2002).
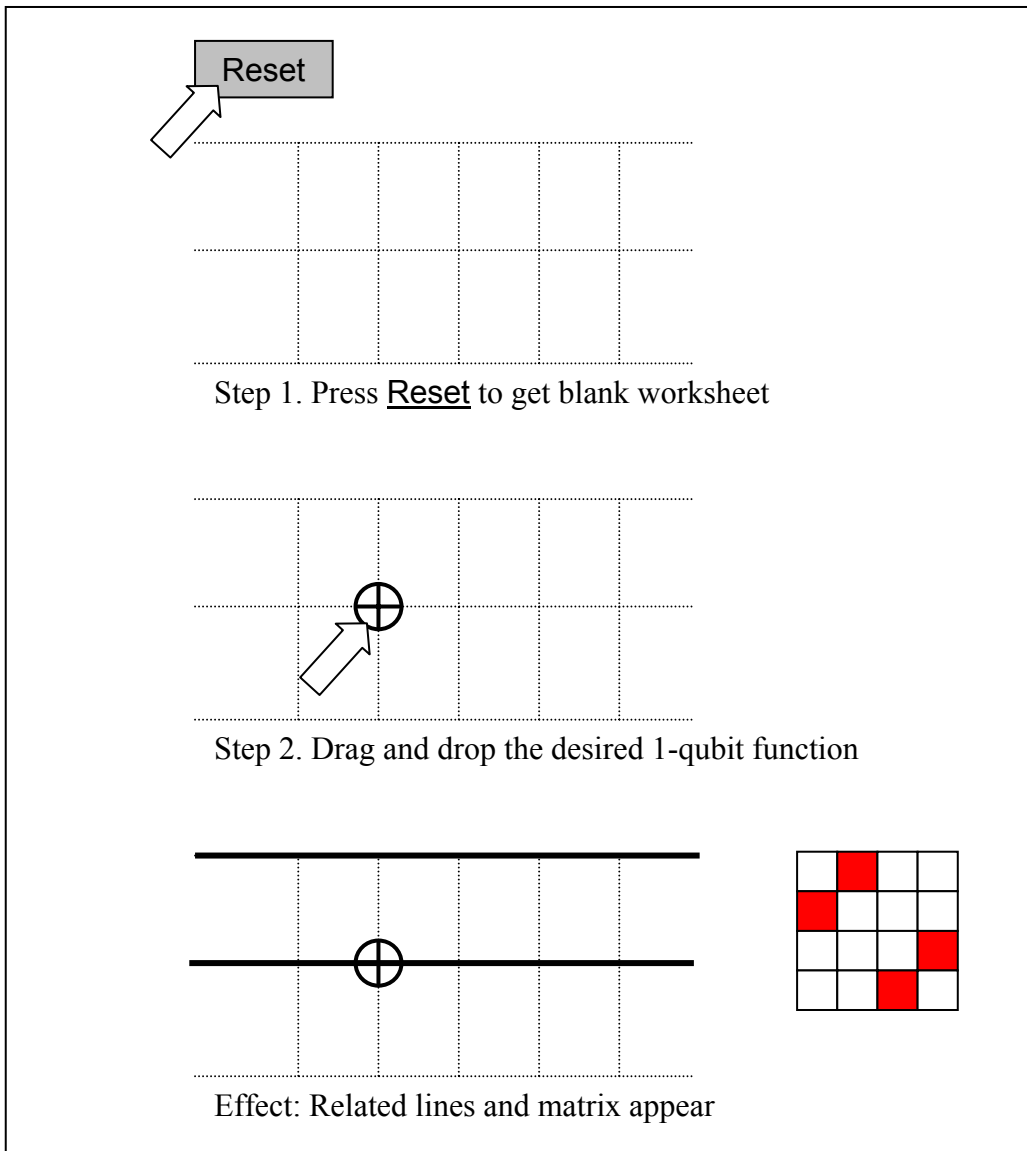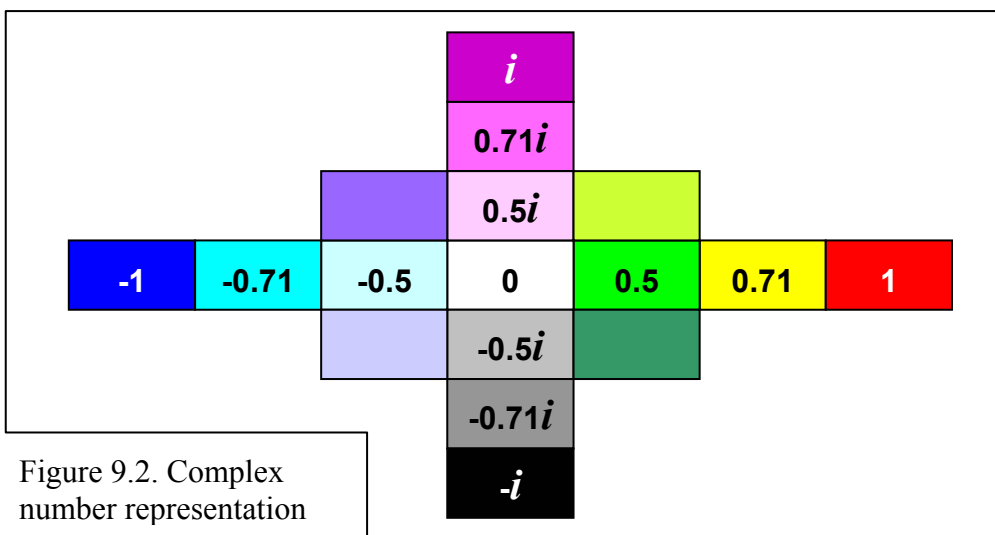
Step 1. Press <u>Reset</u> to get blank worksheet

Step 2. Drag and drop the desired 1-qubit function

Effect: Related lines and matrix appear

Figure 9.1. Quantum Works session. Step 1 amd 2.

			i			
			0.71*i*			
			0.5*i*			
-1	-0.71	-0.5	0	0.5	0.71	1
			-0.5*i*			
			-0.71*i*			
			-*i*			

Figure 9.2. Complex number representation

Step 3. Drag and drop a control

Effect: The connection appears and the matrix changes

Step 4. Drag and drop other 1-qubit function

Effect: New line and new matrix appear
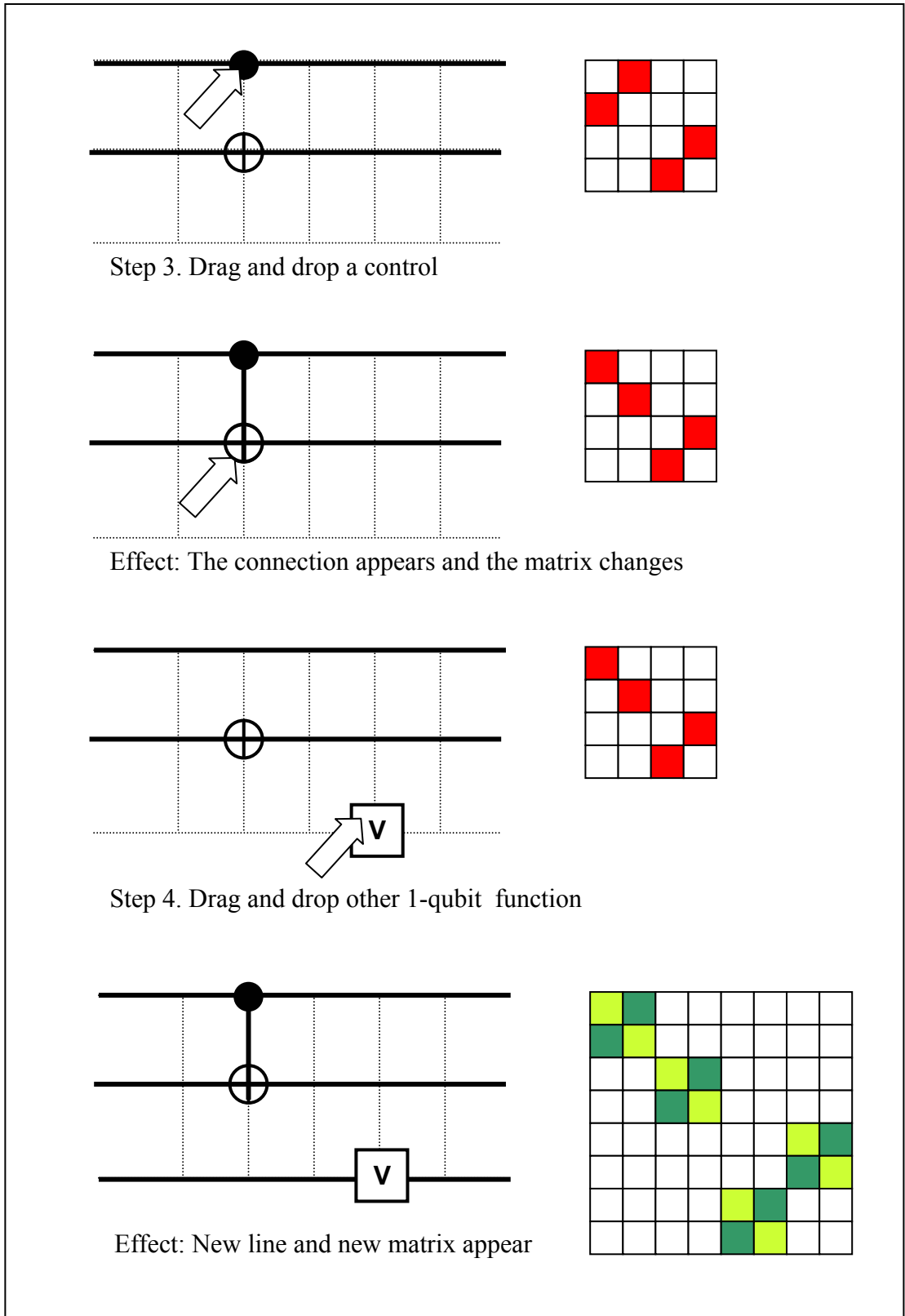
Figure 9.3. Quantum Works session. Step 3 amd 4.

Step 1: Click the content of the matrix for the desired cascade

Synthesize

Step 2: Push **Synthesize** button

Effect: Relevant cascade appears
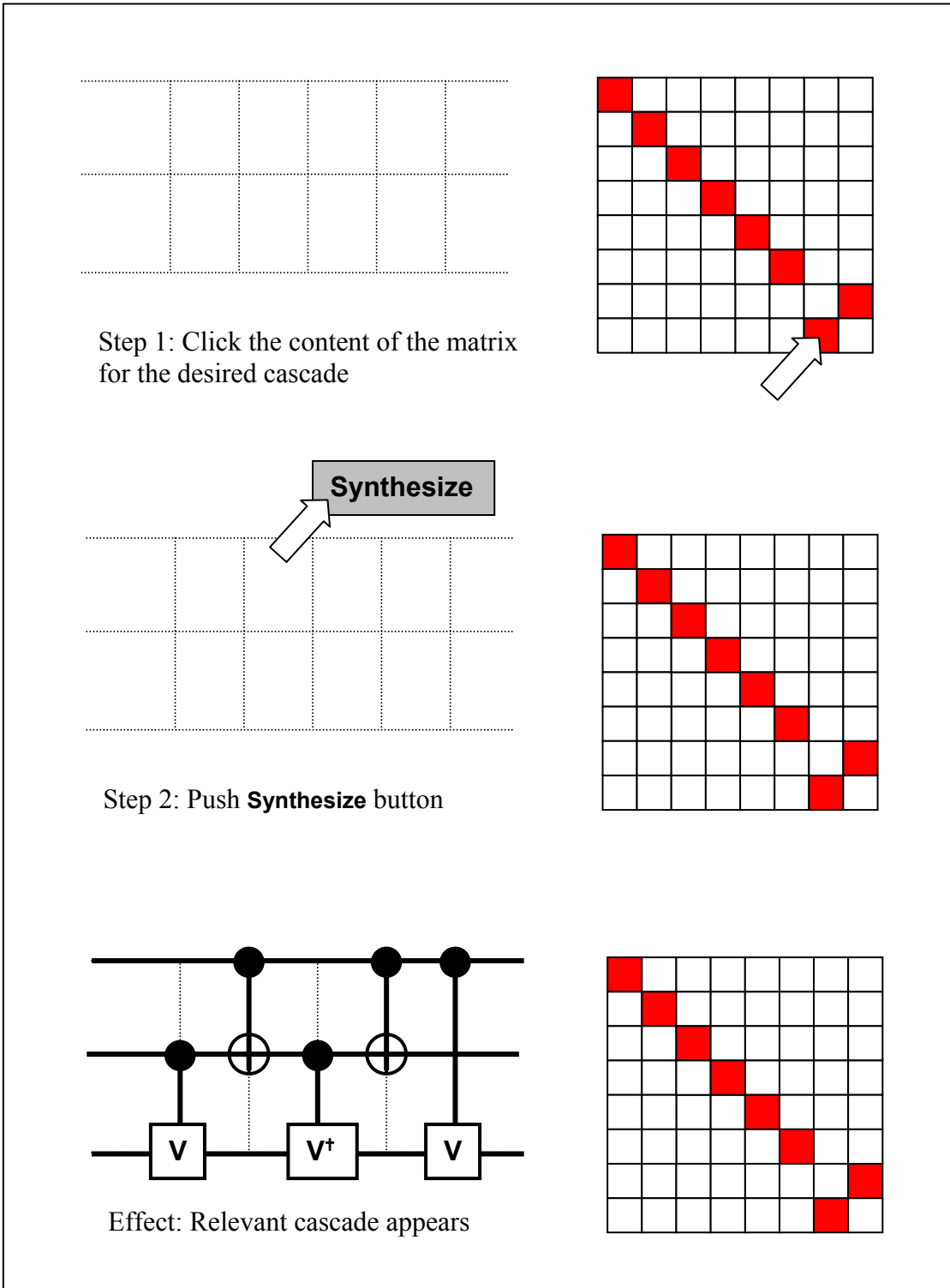
Figure 9.4. Quantum Works session. Automated cascade synthesis

10. Final remarks

This report was intended to make an interdisciplinary reader understand the essence of quantum computing. Hence, several quantum-mechanics-related concepts as Hilbert space, orthonormal bases, spin, Schrödinger equation, etc. were not introduced to not to obstruct the process of knowledge acquisition. It is believed that a reader who has elementary background in programming can now write a simple program for processing states of a 3-qubit register and easily scale it toward operations on four or more qubits.

A challenging task would be to employ the NeuroMaze paradigm (see Liu 2002 www.his.atr.co.jp/ecm/n_maze) to building models of quantum cascades to be run on the ATR's CAM-Brain Machine (CBM). The suggested approach assumes that a given 24×24×24-cell module would be provided with spiketrains representing a 3- or 4-qubit register and return spiketrains representing a single element of a vector of a resulting entangled state. This means that only a single row of a matrix defining a quantum gate would have to be encoded in the module's structure. For, say, 3-qubit register the module's task would be to take get a vector of 8 complex number, multiply each of the number by an appropriate element of the row, and return a representation of the sum of the products.

The book by Mika Hirvensalo (2001) contains the generalized and formalized description of principles of quantum computation, as well as the most impressive quantum algorithms including Grover's Search Algorithm and Shor's Algorithm for Factoring Numbers. The book by Collin P. Williams and Scott H. Clearwater (1998) provides an introduction to quantum computing focusing on quantum mechanics underlying the operations on qubit registers. More about the properties of reversible circuits can be found in (Shende et al. 2002) or (Buller 2003).

Acknowledgements

References

1. Bennett C (1973) Logical reversibility of computation, *IBM Journal of Research and Development, 17*, 525-532.

2. Buller A (2003) Reversible Cascades and 3D Cellular Logic Machine, Technical Report TR-0012, ATR Human Information Science Laboratories, Kyoto.

3. Buller A, Shimohara K (2003) Artificial Mind. Theoretical Background and Research Directions, *Proceedings of the Eighth International Symposium on Artificial Life and Robotics (AROB 8th '03), January 24-26, 2003, Beppu, Oita, Japan*, 506-509.

4. Eccles JC (1994) *How the SELF Controls Its Brain*, Berlin: Springer.

5. Hirvensalo M (2001) *Quantum Computing*, Berlin: Springer.

6. Liu J (2002) *NeuroMaze User's Guide, Version 3.0*, ATR HIS, Kyoto.

7. Lukac M, Perkowski M (2002)Evolving Quantum Circuits Using Genetic Algorithms, *Proceedings, The 4th NASA/DoD Workshop on Evolvable Hardware, July 2002, Washington DC, USA*, 173-181.

8. Lukac M, Pivtoraiko M, Mishchenko A, Perkowski M (2002) Automated Synthesis of Generalized Reversible Cascades using Genetic Algorithms, *Proceedings, 5th International Workshop on Boolean Problems, Freiberg, Germany, September 19-20*, 33-45.

9. Penrose R (1989) *The Emperor's New Mind: Concerning Computers, Minds, and Laws of Physics*, Oxford: Ocford University Press.

10. Shende VV, Prasad AK, Markov IL, Hayes JP (2002) Reversible Logic Circuit Synthesis, *Proceedings, 11th International Workshop on Logic Synthesis*, 125-130.

11. Stapp HP (1993) *Mind, Matter and Quantum Mechanics*, Berlin" Springer.

12. Williams CP & Clearwater SH (1998) *Explorations in Quantum computing*, New York: Springer.