Towards Optical Quantum Computing

Cand.scient thesis in physics
by
Sverre Gullikstad Johnsen

Norwegian University of Science and Technology
NTNU

Department of Physics

Trondheim, Norway

December 2002

Towards Optical Quantum Computing

Sverre Gullikstad Johnsen

December 2002

Imagination is more important than knowledge.

Albert Einstein

To those who taught me.

Preface

This thesis is a principal sketch of a new scheme of quantum computing. The thesis explores some of the fundamentals of quantum optics and tries to put these to work in quantum optical gates. The thesis tries to exploit the properties of classical computing, which has proven quite successful over the last decades, yet leaves the opportunity of parallel processing and other quantum-mechanical effects wide open. In addition to developing quantum optical gates, the thesis expands the present quantum-logic into the realms of trinary algebra. The common Boolean algebraic gates are included as a specialty of the quantum gates proposed. The writing of this thesis has been a plunge into unrevealed terrain, and the work of this thesis may not at all be regarded as fulfilled, however this is a step towards a solution to one of the great concerns of modern computing and quantum computers.

The thesis is divided into four parts.
In Chapter 1, a brief introduction to quantum optics is given, i.e. quantization of the electro-magnetic field, coherent states and the beam-splitter. Some knowledge of the formalism used in quantum mechanics may be required, to fully understand Chapter 1.
In Chapter 2, the history of computing, classical as well as quantum-mechanical, is presented. In addition, the fundamental ideas and concepts of this thesis, regarding optical computing and trinary logic, is presented.
In Chapter 3, a review of classical Boolean algebra and binary logic is given. In addition, a trinary algebra, implementable in quantum-optical computing, is proposed.
In Chapter 4, a variety of quantum optical gates are proposed. Some gates are meant to act as computational devices alone, addition, multiplication, whilst others are meant to perform trinary-logic operations, analogous to the binary-logic operations AND, OR and NOT.

This work was carried out at the Department of Physics at the Norwegian University of Science and Technology under the guidance of Prof. Bo-Sture

K. Skagerstam. Prof. Skagerstam has provided me with a unique source of knowledge on the subject of quantum optics.

I would like to thank my fellow students, Ronny Kjelsberg, Lodve Berre, Øystein O. Langangen, Andreas K. Evensen, Maria V. Bøe, Mona M. Lorentsen, Skjalg A. Fagerjord, Morten Kolstø and Øivind Selnes for a fruitful collaboration over the past year. The discussions concerning theoretical physics, aspects of everyday life and not so common aspects of everyday life as well as technical computer problems have been of great help to me (at least for making the time fly by). Furthermore, I would like to thank Øystein O. Langangen, Ronny Kjelsberg, Liv Gullikstad and Magnar G. Johnsen for reading and providing me with many constructive remarks on the thesis.
Finally, I would like to thank, especially, Bo-Sture K. Skagerstam and all the other lecturers at NTNU for teaching me during my years at the University.

Sverre G. Johnsen

Sverre G. Johnsen
NTNU, Trondheim,
13th of December 2002

Contents

List of Figures

List of Tables

Introduction

The work in this thesis concerns a new scheme of quantum computing. The work is based upon two main ideas;
"It will prove advantageous to expand the present Boolean algebra into the realms of higher order algebra. Since an increase in algebraic order will result in an increase in flow of information."
and
"Quantum-optical, coherent states are extremely resistant to noise and decoherence. They will therefore be very fit for expressing quantum-logical states."

These ideas may be regarded as naive, since we do not discuss them thoroughly. For instance, increased flow of information may be of no interest if the cost of producing multiple-valued logic gates is increasing faster. As we will see, the gates proposed in this thesis also result in intensity loss of the coherent states used, i.e. weakening of the signals, which may actually be regarded as a kind of decoherence.

Both the problems mentioned above, as well as other problems that may arise, will not be discussed in this thesis. Here we merely state a possible solution to the problem of multiple-valued optical quantum computing, and we will leave it to others to verify whether the solution proposed is a good one or not.

Chapter 1

Introduction to Quantum Optics

1.1 Quantization of the Electro-Magnetic Field

Let us consider a classical electro-magnetic field in empty space. In the absence of charges and currents the field satisfies the homogeneous Maxwell equations:

$$\begin{aligned}
\nabla \times \mathbf{E}(\vec{r}, t) &= -\frac{\partial}{\partial t}\mathbf{B}(\vec{r}, t) \ , && (1.1)\\
\nabla \times \mathbf{B}(\vec{r}, t) &= \frac{1}{c^2}\frac{\partial}{\partial t}\mathbf{E}(\vec{r}, t) \ , && (1.2)\\
\nabla \cdot \mathbf{E}(\vec{r}, t) &= 0 \ , && (1.3)\\
\nabla \cdot \mathbf{B}(\vec{r}, t) &= 0 \ , && (1.4)
\end{aligned}$$

where $\mathbf{E}(\vec{r}, t)$ and $\mathbf{B}(\vec{r}, t)$ are the usual electric and magnetic field-vectors at the space-time point $(\vec{r}, t)$. We will represent the free electro-magnetic field by the transverse vector potential $\mathbf{A}(\vec{r}, t)$, which satisfies the homogeneous wave equation

$$\nabla^2 \mathbf{A}(\vec{r}, t) - \frac{1}{c^2}\frac{\partial^2}{\partial t^2}\mathbf{A}(\vec{r}, t) = 0 \ , \tag{1.5}$$

and the radiation gauge condition

$$\nabla \cdot \mathbf{A}(\vec{r}, t) = 0 \ . \tag{1.6}$$

The electric and the magnetic fields are given, in terms of $\mathbf{A}(\vec{r},t)$, by the equations

$$\mathbf{E}(\vec{r},t) = -\frac{\partial}{\partial t}\mathbf{A}(\vec{r},t) , \tag{1.7}$$
$$\mathbf{B}(\vec{r},t) = \nabla \times \mathbf{A}(\vec{r},t) . \tag{1.8}$$

1.1.1 Plane wave expansions

To obtain the Hamiltonian equations of motion, we shall make a Fourier decomposition of $\mathbf{A}(\vec{r},t)$ with respect to it's space variables x, y and z. We imagine the electro-magnetic field to be contained in a large box of sides L, and we impose periodic boundaries on the field.

We now write the three-dimensional Fourier expansion of $\mathbf{A}(\vec{r},t)$ in terms of plane-wave modes in the form

$$\mathbf{A}(\vec{r},t) = \frac{1}{\epsilon_0^{1/2}L^{3/2}}\sum_k \mathcal{A}_{\mathbf{k}}(t)e^{i\mathbf{k}\cdot\mathbf{r}} , \tag{1.9}$$

where the vector $\mathbf{k}$ has components

$$\left.\begin{array}{ll} k_1 = 2\pi n_1/L , & n_1 = 0, \pm 1, \pm 2, ... \\ k_2 = 2\pi n_2/L , & n_2 = 0, \pm 1, \pm 2, ... \\ k_3 = 2\pi n_3/L , & n_3 = 0, \pm 1, \pm 2, ... \end{array}\right\} \tag{1.10}$$

forming a discrete set. The sum $\sum_{\mathbf{k}}$ is understood to be a sum over the integers n_1, n_2, n_3. The factor $\frac{1}{\epsilon_0^{1/2}L^{3/2}}$ where ϵ_0 is the vacuum dielectric constant, is introduced for later convenience.

If we apply Eq. (1.6) to Eq. (1.9), we easily see that we get the condition

$$\mathbf{k}\cdot\mathcal{A}_{\mathbf{k}}(t) = 0 . \tag{1.11}$$

In addition, the reality of $\mathbf{A}(\vec{r},t)$ leads to the condition

$$\mathcal{A}_{-\mathbf{k}}(t) = \mathcal{A}_{\mathbf{k}}^*(t) . \tag{1.12}$$

Since we have Eq. (1.5), it follows immediately that $\mathcal{A}_{\mathbf{k}}(t)$ satisfies the equation of motion

$$\left(\frac{\partial^2}{\partial t^2} + \omega^2\right)\mathcal{A}_{\mathbf{k}}(t) = 0 , \tag{1.13}$$

where we have introduced the angular frequency $\omega_k = ck$ and abbreviated it by writing ω. The general solution to this equation, which also obeys the condition (1.12), is given by

$$\mathcal{A}_{\mathbf{k}}(t) = \mathbf{c}_{\mathbf{k}} e^{-i\omega t} + \mathbf{c}^*_{-\mathbf{k}} e^{i\omega t} \ . \tag{1.14}$$

1.1.2 Unit polarization vectors

It will prove advantageous to resolve the vector $\mathbf{c}_{\mathbf{k}}$ into two orthogonal components, which are chosen such that Eq. (1.11) is satisfied automatically. This is most easily done by choosing a pair of orthonormal real base vectors $\epsilon_{\mathbf{k}1}, \epsilon_{\mathbf{k}2}$ that obey the conditions

$$\left.\begin{array}{ll} \mathbf{k}\cdot\epsilon_{\mathbf{k}s} = 0 \ , & (s = 1,2) \\ \epsilon^*_{\mathbf{k}s}\cdot\epsilon_{\mathbf{k}s'} = \delta_{ss'} \ , & (s,s' = 1,2) \\ \epsilon_{\mathbf{k}1}\times\epsilon_{\mathbf{k}2} = \mathbf{k}/k \equiv \kappa \ , & \end{array}\right\} \tag{1.15}$$

which signify transversality, orthonormality and right-handedness, respectively, and then putting

$$\mathbf{c}_{\mathbf{k}} = \sum_{s=1}^{2} c_{\mathbf{k}s}\epsilon_{\mathbf{k}s} \ . \tag{1.16}$$

On substituting from Eq. (1.16) into Eq. (1.14) and using the result from Eq. (1.9) we get

$$\begin{aligned} \mathbf{A}(\mathbf{r},t) &= \frac{1}{\epsilon_0^{1/2}L^{3/2}}\sum_{\mathbf{k}}\sum_{s}[c_{\mathbf{k}s}\epsilon_{\mathbf{k}s}e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)} + c^*_{\mathbf{k}s}\epsilon^*_{\mathbf{k}s}e^{-i(\mathbf{k}\cdot\mathbf{r}-\omega t)}] \\ &= \frac{1}{\epsilon_0^{1/2}L^{3/2}}\sum_{\mathbf{k}}\sum_{s}[u_{\mathbf{k}s}(t)\epsilon_{\mathbf{k}s}e^{i\mathbf{k}\cdot\mathbf{r}} + u^*_{\mathbf{k}s}(t)\epsilon^*_{\mathbf{k}s}e^{-i\mathbf{k}\cdot\mathbf{r}}] \ , \end{aligned} \tag{1.17}$$

where we have written

$$u_{\mathbf{k}s}(t) = c_{\mathbf{k}s}e^{-i\omega t} \ . \tag{1.18}$$

We may now make use of Eq. (1.7) and Eq. (1.8) to write mode expansions for the $\mathbf{E}(\vec{r},t)$ and $\mathbf{B}(\vec{r},t)$ vectors. Thus

$$\mathbf{E}(\vec{r},t) = \frac{i}{\epsilon_0^{1/2}L^{3/2}}\sum_{\mathbf{k}}\sum_{s}\omega[u_{\mathbf{k}s}(t)\epsilon_{\mathbf{k}s}e^{i\mathbf{k}\cdot\mathbf{r}} - c.c.] \ , \tag{1.19}$$

and

$$\mathbf{B}(\vec{r},t) = \frac{i}{\epsilon_0^{1/2}L^{3/2}}\sum_{\mathbf{k}}\sum_{s}[u_{\mathbf{k}s}(t)(\mathbf{k}\times\epsilon_{\mathbf{k}s})e^{i\mathbf{k}\cdot\mathbf{r}} - c.c.] \ . \tag{1.20}$$

1.1.3 Energy of the electro-magnetic field

The energy of the field is given by

$$H = \frac{1}{2}\int_{L^3}\left[\epsilon_0 \mathbf{E}^2(\vec{r},t) + \frac{1}{\mu_0}\mathbf{B}^2(\vec{r},t)\right] d^3r \ . \tag{1.21}$$

When solving this expression we obtain

$$H = 2\sum_{\mathbf{k}}\sum_{s}\omega^2 |u_{\mathbf{k}s}(t)|^2 \ , \tag{1.22}$$

which expresses the energy as a sum over the modes.

For the purpose of field quantization it is convenient to write H in Hamiltonian form, which we do by introducing a pair of real canonical variables $q_{\mathbf{k}s}(t)$ and $p_{\mathbf{k}s}(t)$ defined by

$$q_{\mathbf{k}s}(t) = [u_{\mathbf{k}s}(t) + u^*_{\mathbf{k}s}(t)] \ , \tag{1.23}$$

$$p_{\mathbf{k}s}(t) = -i\omega[u_{\mathbf{k}s}(t) - u^*_{\mathbf{k}s}(t)] \ . \tag{1.24}$$

Substitution of Eq. (1.23) and Eq. (1.24) in Eq. (1.22) yields

$$H = \frac{1}{2}\sum_{\mathbf{k}}\sum_{s}[p^2_{\mathbf{k}s}(t) + \omega^2 q^2_{\mathbf{k}s}(t)] \ . \tag{1.25}$$

This may be recognized as a system of independent harmonic oscillators, one for each $\mathbf{k}, s$ mode of the electro-magnetic field.

1.1.4 Canonical quantization of the transverse field

In order to describe the electro-magnetic field in quantum mechanics, we have to associate Hilbert space operators with the dynamical variables. The Hilbert space operators will be denoted by the same symbols as their dynamical counterparts, but with the caret $\hat{}$; e.g. the operators corresponding to $q_{\mathbf{k}s}(t)$ and $p_{\mathbf{k}s}(t)$ will be denoted by $\hat{q}_{\mathbf{k}s}(t)$ and $\hat{p}_{\mathbf{k}s}(t)$. As the classical variables associated with two different modes are uncoupled, the corresponding Hilbert space operators commute. We therefore have

$$[\hat{q}_{\mathbf{k}s}(t), \hat{p}_{\mathbf{k}'s'}(t)] = i\hbar\delta^3_{\mathbf{k}\mathbf{k}'}\delta_{ss'} \ , \tag{1.26}$$

$$[\hat{q}_{\mathbf{k}s}(t), \hat{q}_{\mathbf{k}'s'}(t)] = 0 \ , \tag{1.27}$$

$$[\hat{p}_{\mathbf{k}s}(t), \hat{p}_{\mathbf{k}'s'}(t)] = 0 \ . \tag{1.28}$$

We may now write the quantum mechanical Hamiltonian

$$\hat{H} = \frac{1}{2}\sum_{\mathbf{k}}\sum_{s}[\hat{p}^2_{\mathbf{k}s}(t) + \omega^2\hat{q}^2_{\mathbf{k}s}(t)] \ . \tag{1.29}$$

For our purposes it will be more convenient to deal with a set of non-Hermitian operators defined by

$$\begin{aligned} \hat{a}_{\mathbf{k}s}(t) &= \frac{1}{(2\hbar\omega)^{1/2}}[\omega\hat{q}_{\mathbf{k}s}(t) + i\hat{p}_{\mathbf{k}s}(t)] \ , & (1.30)\\ \hat{a}^\dagger_{\mathbf{k}s}(t) &= \frac{1}{(2\hbar\omega)^{1/2}}[\omega\hat{q}_{\mathbf{k}s}(t) - i\hat{p}_{\mathbf{k}s}(t)] \ , & (1.31) \end{aligned}$$

where the second is the Hermitian conjugate of the first. If inverted we get the relations

$$\begin{aligned} \hat{q}_{\mathbf{k}s}(t) &= (\hbar/2\omega)^{1/2}[\hat{a}_{\mathbf{k}s}(t) + \hat{a}^\dagger_{\mathbf{k}s}(t)] \ , & (1.32)\\ \hat{p}_{\mathbf{k}s}(t) &= i(\hbar\omega/2)^{1/2}[\hat{a}^\dagger_{\mathbf{k}s}(t) - \hat{a}_{\mathbf{k}s}(t)] \ . & (1.33) \end{aligned}$$

The commutation relations for $\hat{a}_{\mathbf{k}s}(t)$ and $\hat{a}^\dagger_{\mathbf{k}s}(t)$ become

$$\begin{aligned} [a_{\mathbf{k}s}(t), a^\dagger_{\mathbf{k}'s'}(t)] &= \delta^3_{\mathbf{k}\mathbf{k}'}\delta_{ss'} \ , & (1.34)\\ [a_{\mathbf{k}s}(t), a_{\mathbf{k}'s'}(t)] &= 0 \ , & (1.35)\\ [a^\dagger_{\mathbf{k}s}(t), a^\dagger_{\mathbf{k}'s'}(t)] &= 0 \ . & (1.36) \end{aligned}$$

It is evident that apart from some scaling factor, $\hat{a}_{\mathbf{k}s}(t)$ and $\hat{a}^\dagger_{\mathbf{k}s}(t)$ correspond with the complex amplitudes $u_{\mathbf{k}s}(t)$ and $u^*_{\mathbf{k}s}(t)$, and they also have the same time dependence,

$$\begin{aligned} \hat{a}_{\mathbf{k}s}(t) &= \hat{a}_{\mathbf{k}s}(0)e^{-i\omega t} \ , & (1.37)\\ \hat{a}^\dagger_{\mathbf{k}s}(t) &= \hat{a}^\dagger_{\mathbf{k}s}(0)e^{i\omega t} \ . & (1.38) \end{aligned}$$

As we clearly see, the operator products $\hat{a}_{\mathbf{k}s}(t)\hat{a}^\dagger_{\mathbf{k}s}(t)$ and $\hat{a}^\dagger_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t)$ are time independent. Let us substitute for $\hat{q}_{\mathbf{k}s}(t)$ and $\hat{p}_{\mathbf{k}s}(t)$ in the Hamiltonian form of Eq. (1.29). The Hamiltonian now becomes

$$\hat{H} = \frac{1}{2}\sum_{\mathbf{k}}\sum_{s}\hbar\omega[\hat{a}_{\mathbf{k}s}(t)\hat{a}^\dagger_{\mathbf{k}s}(t) + \hat{a}^\dagger_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t)] \ . \tag{1.39}$$

Alternatively

$$\hat{H} = \sum_{\mathbf{k}}\sum_{s}\hbar\omega[\hat{a}^\dagger_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t) + \frac{1}{2}] \ , \tag{1.40}$$

if we make use of Eq. (1.34). We notice that the so-called *zero point contribution*, $\frac{\hbar\omega}{2}$, contributes with an infinite energy for an unbounded set of modes. This is a difficulty of QED that has never been resolved satisfactorily (Ref. [1]). One may argue that this contribution has no physical meaning , so we will merely neglect the term and write

$$\hat{H} = \sum_{\mathbf{k}} \sum_{s} \hbar\omega \hat{a}^{\dagger}_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t) \ , \tag{1.41}$$

for the energy.

1.1.5 Spectrum of the energy; photons

The Hermitian operator $\hat{a}^{\dagger}_{\mathbf{k}s}\hat{a}_{\mathbf{k}s}$ that appears in Eq. (1.41) is a particularly important one and will be denoted by $\hat{n}_{\mathbf{k}s}$. It is called the *number operator.* We will denote the eigenvalues of the number operator by $n_{\mathbf{k}s}$ and the eigenstates by $|n_{\mathbf{k}s}\rangle$, such that

$$\hat{n}_{\mathbf{k}s} \ |n_{\mathbf{k}s}\rangle = n_{\mathbf{k}s} \ |n_{\mathbf{k}s}\rangle \ . \tag{1.42}$$

Since the number operator is self-adjoint, the eigenvalues are real, and it can be shown that they are also non-negative. An important difference between the classical energy expression in Eq. (1.21) and the quantum mechanical energy operator in Eq. (1.41) is the energy spectrum. In the classical case the energy may take any non-negative value. In the quantum mechanical case, however, only discrete, non-negative, quantized values may be admitted as eigenvalues of the energy operator[1].

The electro-magnetic field will be represented by a Hilbert space vector

$$\begin{aligned} |\phi\rangle &= \prod_{\mathbf{k}s} |n_{\mathbf{k}s}\rangle & (1.43) \\ &= |n_{\mathbf{k_1}s_1}\rangle |n_{\mathbf{k_2}s_2}\rangle ... & (1.44) \\ &= |\{n\}\rangle. & (1.45) \end{aligned}$$

Such a state is known as a Fock state and is characterized by the set of occupation numbers $n_{\mathbf{k_1}s_1}, n_{\mathbf{k_2}s_2}, ...$ for all the modes. It follows that the electro-magnetic field states are eigenstates of the Hamilton operator given by Eq. (1.41), such that

$$\hat{H} \ |\{n\}\rangle = \big(\sum_{\mathbf{k},s} n_{\mathbf{k}s}\hbar\omega\big)|\{n\}\rangle \ . \tag{1.46}$$

[1]Since $\hat{n}_{\mathbf{k}s}$ contains squares of the operators $\hat{p}$ and $\hat{x}$ only, the expectation value of $\hat{n}_{\mathbf{k}s}$ in any state cannot be negative, and hence the eigenvalues of $\hat{n}_{\mathbf{k}s}$ must be non-negative.

The discrete excitations, or quanta, of the electro-magnetic field, corresponding to the occupation numbers n, are usually known as *photons*. Thus a state $|..., 0, 0, 1_{\mathbf{k}s}, 0, 0, ...\rangle$ is described as a state with one photon of wave vector $\mathbf{k}$ and polarization s. We may later find it convenient to denote this state as $|1\rangle_{\mathbf{k}s}$.

$\hat{a}_{\mathbf{k}s}(t)$ and $\hat{a}^\dagger_{\mathbf{k}s}(t)$ are known as the *annihilation* and *creation* operators respectively. The reason for this is that they lower and raise the photon number corresponding to $\mathbf{k}$ and s by one. This can be realized from the following proof.

Proof. First, construct the state $\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle$.
Let us then operate on the state by the number operator:

$$\begin{aligned}\hat{n}_{\mathbf{k}s}\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle &= \hat{a}^\dagger_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle \\ &= \left(\hat{a}_{\mathbf{k}s}(t)\hat{a}^\dagger_{\mathbf{k}s}(t) - 1\right)\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle \\ &= (n-1)\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle\ , \end{aligned} \tag{1.47}$$

when applying Eq. (1.34).
Thus $\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle$ is an eigenstate of the number operator and may when applying Eq. (1.42), be written as

$$\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle = c_a|n_{\mathbf{k}s} - 1\rangle\ , \tag{1.48}$$

where c_a is a normalization constant.
Futhermore, we have

$$\begin{aligned}\langle n_{\mathbf{k}s} - 1|n_{\mathbf{k}s} - 1\rangle &= \frac{1}{c_a^2}\langle n_{\mathbf{k}s}|\hat{a}^\dagger_{\mathbf{k}s}(t)\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle \\ &= \frac{n_{\mathbf{k}s}}{c_a^2}\cdot\langle n_{\mathbf{k}s}|n_{\mathbf{k}s}\rangle\ , \end{aligned} \tag{1.49}$$

and by normalization, we get

$$c_a = \sqrt{n}_{\mathbf{k}s}\ , \tag{1.50}$$

and

$$\hat{a}_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle = \sqrt{n}_{\mathbf{k}s}|n_{\mathbf{k}s} - 1\rangle\ . \tag{1.51}$$

It may by following the same procedure, be shown that

$$\hat{a}^\dagger_{\mathbf{k}s}(t)|n_{\mathbf{k}s}\rangle = \sqrt{n_{\mathbf{k}s} + 1}\cdot|n_{\mathbf{k}s} + 1\rangle\ . \tag{1.52}$$

$\square$

The boson properties of photons are easily realized since the eigenvalues of the number operators are unbounded and there may be found an arbitrary number of photons in the same state.

1.2 Coherent States

Coherent states can be shown to be near-classical quantum states. The coherent states of the field comes as close as possible to being classical states of definite complex amplitude (Ref. [1]). It is known that coherent states are particularly appropriate for the description of lasers and parametric oscillators.

1.2.1 Fock representation of the coherent state

In this section we will focus on one single mode of the electro-magnetic field. The theory may easily be expanded to cover a multiple mode state. We shall therefore simplify the notation by discarding the mode label, $\mathbf{k}, s$.

We will make use of the properties of the annihilation and creation operators, namely

$$\begin{aligned} \hat{a}|n\rangle &= \sqrt{n}\ |n-1\rangle\ , && (1.53) \\ \hat{a}^{\dagger}|n\rangle &= \sqrt{n+1}\ |n+1\rangle\ . && (1.54) \end{aligned}$$

It may be shown that $n \geq 0$ so that $\hat{a}|0\rangle \equiv 0$.

Assume that there exists an eigenstate of $\hat{a}$, $|\alpha\rangle$, such that

$$\hat{a}|\alpha\rangle = \alpha|\alpha\rangle\ . \tag{1.55}$$

The state $|\alpha\rangle$ is for many reasons known as a coherent state (Ref. [1]). Since the Fock states form a complete set, $|\alpha\rangle$ may be represented as

$$|\alpha\rangle = \sum_{n=0}^{\infty} c_n |n\rangle\ , \tag{1.56}$$

in which c_n are complex numbers. On substituting Eq. (1.56) in Eq. (1.55) and by using Eq. (1.53) we get

$$\begin{aligned} \hat{a}|\alpha\rangle &= \sum_{n=1}^{\infty} c_n \sqrt{n} |n-1\rangle \\ &= \alpha \sum_{n=0}^{\infty} c_n |n\rangle\ . \end{aligned}$$

As the $|n\rangle$ (n=(0,1,2,...)) form a set of orthogonal state vectors, this equation is satisfied if and only if the coefficients of the corresponding Fock vectors on both sides are equal. We therefore have

$$c_n = \frac{\alpha}{\sqrt{n}}\ c_{n-1}\ . \tag{1.57}$$

By repeated application we obtain

$$c_n = \frac{\alpha^2}{\sqrt{n(n-1)}}\ c_{n-2} = ... = \frac{\alpha^n}{\sqrt{(n!)}}\ c_0\ , \tag{1.58}$$

so that

$$|\alpha\rangle = c_0 \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{(n!)}}\ |n\rangle\ . \tag{1.59}$$

If we require $\langle\alpha|\alpha\rangle \equiv 1$, we get that

$$c_0 = e^{-|\alpha|^2/2}\ , \tag{1.60}$$

and

$$|\alpha\rangle = e^{-|\alpha|^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{(n!)}}\ |n\rangle\ . \tag{1.61}$$

When $\alpha = 0$, the coherent state becomes the vacuum state $|0\rangle$, which is both a coherent state and a Fock state. The probability of finding n photons in a coherent state, is given by the projection

$$\begin{aligned} p_\alpha(n) &= |\langle n|\alpha\rangle|^2 \\ &= |e^{-|\alpha|^2/2}\frac{\alpha^n}{\sqrt{(n!)}}|^2 \\ &= \frac{|\alpha|^{2n}}{n!}e^{-|\alpha|^2}\ , \end{aligned} \tag{1.62}$$

which we recognize as the Poisson distribution in n. The mean number of photons in a coherent state is given by

$$\sum_{n=0}^{\infty} n p_\alpha(n) = |\alpha|^2 = \langle\alpha|\hat{n}|\alpha\rangle\ . \tag{1.63}$$

A remarkable feature of a coherent state is that it is not altered by the annihilation operator, and the number of photons at any time is totally random. Even for extremely small αs there is a slight probability of finding a huge number of photons.

1.2.2 The displacement operator

By making use of Eq. (1.61) and the properties of the creation operator (Eq. (1.54)), we may write

$$
\begin{aligned}
|\alpha\rangle &= e^{-|\alpha|^2/2}\sum_{n=0}^{\infty}\frac{\alpha^n \hat{a}^{\dagger n}}{n!}|0\rangle \\
&= e^{-|\alpha|^2/2}e^{\alpha\hat{a}^\dagger}|0\rangle \ ,
\end{aligned}
\tag{1.64}
$$

which shows that a coherent state is to be regarded as a displaced vacuum state. We may express this in a somewhat more symmetric way by making use of the properties of the annihilation operator (Eq. (1.53)). Thus

$$
|\alpha\rangle = e^{-|\alpha|^2/2}e^{\alpha\hat{a}^\dagger}e^{-\alpha^*\hat{a}}|0\rangle \ . \tag{1.65}
$$

We now make use of the Campbell-Baker-Hausdorff operator-identity (Ref. [1]) for two operators $\hat{A}, \hat{B}$;

$$
e^{\hat{A}+\hat{B}} = e^{\hat{A}}\ e^{\hat{B}}\ e^{-[\hat{A},\hat{B}]/2} \ , \tag{1.66}
$$

provided that

$$
[\hat{A},[\hat{A},\hat{B}]] = 0 = [\hat{B},[\hat{A},\hat{B}]] \ . \tag{1.67}
$$

It can be proven that Eq. (1.67) holds for the creation and annihilation operators. It can easily be checked that we may now write Eq. (1.65) as

$$
|\alpha\rangle = \hat{D}_a(\alpha)|0\rangle = e^{\alpha\hat{a}^\dagger - \alpha^*\hat{a}}|0\rangle \ . \tag{1.68}
$$

$\hat{D}_a(\alpha)$ is the *displacement operator*, which creates a coherent state $|\alpha\rangle$ from the vacuum state $|0\rangle$.

1.2.3 Coherent states and orthogonality

One important feature of the coherent states is their near-orthonormality property, i.e.

$$
\langle\beta|\alpha\rangle \simeq \begin{cases} 1 & \alpha = \beta \ , \\ 0 & \alpha \neq \beta \ , \end{cases} \tag{1.69}
$$

wich we sloppily may rewrite,

$$
\langle\beta|\alpha\rangle \simeq \delta_{\alpha,\beta} \ . \tag{1.70}
$$

This can be seen from Eq. (1.61),

$$
\begin{aligned}
\langle\beta|\alpha\rangle &= \Big(e^{-|\beta|^2/2}\sum_{m=0}^{\infty}\frac{\beta^{*m}}{\sqrt{(m!)}}\langle m|\Big)\cdot\Big(e^{-|\alpha|^2/2}\sum_{n=0}^{\infty}\frac{\alpha^{n}}{\sqrt{(n!)}}|n\rangle\Big) \\
&= e^{-\frac{(|\alpha|^2+|\beta|^2)}{2}}\sum_{m,n=0}^{\infty}\frac{\beta^{*m}\alpha^n}{\sqrt{m!n!}}\underbrace{\langle m|n\rangle}_{\delta_{m,n}} \\
&= e^{-\frac{(|\alpha|^2+|\beta|^2)}{2}}\sum_{n=0}^{\infty}\frac{\beta^{*n}\alpha^n}{n!} \\
&= e^{-\frac{(|\alpha|^2+|\beta|^2)}{2}}e^{\beta^*\alpha}\ , \qquad (1.71)
\end{aligned}
$$

$$
\begin{aligned}
|\langle\beta|\alpha\rangle|^2 &= \exp\left(-|\alpha|^2-|\beta|^2+\beta^*\alpha+\beta\alpha^*\right) \\
&= \exp\left(-|\alpha-\beta|^2\right)\ . \qquad (1.72)
\end{aligned}
$$

Assuming $\alpha,\beta\to\infty$, we realize that Eq. (1.69) holds.
This means that the coherent states are near-orthonormal, i.e. for large αs (and βs) the cross terms will disappear,

$$\left(c_\alpha^*\langle\alpha|+c_\beta^*\langle\beta|\right)\left(c_\alpha|\alpha\rangle+c_\beta|\beta\rangle\right)\simeq|c_\alpha|^2+|c_\beta|^2\stackrel{!}{=}1\ . \qquad (1.73)$$

1.2.4 Coherent states and the beam-splitter

The beam-splitter is a semi-transparent mirror where the transmittivity is given by

$$T=\eta\ , \qquad (1.74)$$

and the reflectivity is given by

$$R=1-T=1-\eta\ . \qquad (1.75)$$

In quantum optics the beam-splitter has two input channels and two output channels. In classical optics, input channels which experience no intensity may be ignored. In the quantum optical case, however, we must consider both input channels at all times. The vacuum state (origin of no intensity) is also a state of the electro-magnetic field and may interfere with the other input-state. Assuming there is a linear relation between $\hat{1}$,$\hat{2}$,$\hat{a}$ and $\hat{b}$, the output annihilation operators, $\hat{a},\hat{b}$, are given by

$$
\begin{aligned}
\hat{a} &= \sqrt{\eta}\ \hat{1}+\sqrt{1-\eta}\ \hat{2}\ , \qquad (1.76) \\
\hat{b} &= \sqrt{1-\eta}\ \hat{1}-\sqrt{\eta}\ \hat{2}\ , \qquad (1.77)
\end{aligned}
$$

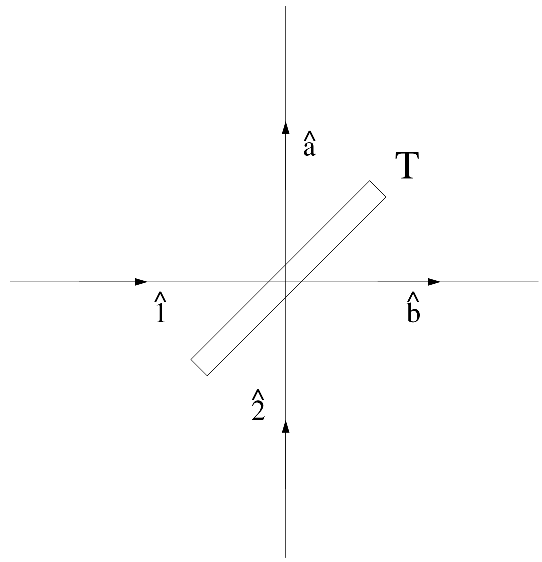


Figure 1.1: Schematic drawing of a beam-splitter with transmittivity T, input operators $\hat{1}$, $\hat{2}$ and output operators $\hat{a}$, $\hat{b}$.

which satisfy the commutation relations in Eq. (1.34)-(1.36). When inverted these relations give

$$(\sqrt{\eta}+\sqrt{1-\eta})(\hat{a}+\hat{b}) = (1+2\sqrt{\eta}\sqrt{1-\eta})\hat{1}+(1-2\eta)\hat{2} \ , \tag{1.78}$$

$$(\sqrt{\eta}+\sqrt{1-\eta})(\hat{a}-\hat{b}) = (2\eta-1)\hat{1}+(1+2\sqrt{\eta}\sqrt{1-\eta})\hat{2} \ . \tag{1.79}$$

In the special case, which will be most important to us, where $T=1/2$, we get

$$\hat{1} = \sqrt{\frac{1}{2}}(\hat{a}+\hat{b}) \ , \tag{1.80}$$

$$\hat{2} = \sqrt{\frac{1}{2}}(\hat{a}-\hat{b}) \ . \tag{1.81}$$

Furthermore, in the case where $T = 1/2$, the displacement operators become

$$\begin{aligned}
\hat{D}_1(\alpha) &= e^{\alpha\hat{1}^\dagger - \alpha^*\hat{1}} \\
&= \exp\left[\alpha\sqrt{\frac{1}{2}}(\hat{a}^\dagger + \hat{b}^\dagger) - \alpha^*\sqrt{\frac{1}{2}}(\hat{a} + \hat{b})\right] \\
&= \hat{D}_a(\frac{\alpha}{\sqrt{2}})\hat{D}_b(\frac{\alpha}{\sqrt{2}}) \ , \qquad (1.82)
\end{aligned}$$

$$\begin{aligned}
\hat{D}_2(\alpha) &= e^{\alpha\hat{2}^\dagger - \alpha^*\hat{2}} \\
&= \exp\left[\alpha\sqrt{\frac{1}{2}}(\hat{a}^\dagger - \hat{b}^\dagger) - \alpha^*\sqrt{\frac{1}{2}}(\hat{a} - \hat{b})\right] \\
&= \hat{D}_a(\frac{\alpha}{\sqrt{2}})\hat{D}_b(-\frac{\alpha}{\sqrt{2}}) \ . \qquad (1.83)
\end{aligned}$$

If the two input states are $|\alpha\rangle$ and $|\beta\rangle$, we experience interference and the output states will be

$$\begin{aligned}
|input\rangle_{1,2} &= |\alpha\rangle_1|\beta\rangle_2 \\
&= D_1(\alpha)D_2(\beta)|0\rangle \qquad (1.84) \\
|output\rangle_{a,b} &= \hat{D}_a(\frac{\alpha}{\sqrt{2}})\hat{D}_b(\frac{\alpha}{\sqrt{2}})\hat{D}_a(\frac{\beta}{\sqrt{2}})\hat{D}_b(-\frac{\beta}{\sqrt{2}})|0\rangle \ . \qquad (1.85)
\end{aligned}$$

And since $\hat{a}$ and $\hat{b}$ commute, we may write

$$\begin{aligned}
&\hat{D}_a(\frac{\alpha}{\sqrt{2}})\hat{D}_b(\frac{\alpha}{\sqrt{2}})\hat{D}_a(\frac{\beta}{\sqrt{2}})\hat{D}_b(-\frac{\beta}{\sqrt{2}})|0\rangle = \\
&\hat{D}_a(\frac{\alpha}{\sqrt{2}})\hat{D}_a(\frac{\beta}{\sqrt{2}})\hat{D}_b(\frac{\alpha}{\sqrt{2}})\hat{D}_b(-\frac{\beta}{\sqrt{2}})|0\rangle = \\
&\qquad \hat{D}_a(\frac{\alpha+\beta}{\sqrt{2}})\hat{D}_b(\frac{\alpha-\beta}{\sqrt{2}})|0\rangle \ . \qquad (1.86)
\end{aligned}$$

Chapter 2

Introduction to Trinary Quantum Computing

2.1 Quantum Computing, So Far

2.1.1 History of classical and quantum computing

Quantum computation and quantum information is the study of the information processing tasks that can be accomplished using quantum mechanical systems. Quantum mechanics is a mathematical framework or set of rules for the construction of physical theories. One of the goals of quantum computation and quantum information is to develop tools which sharpen our intuition about quantum mechanics, and make its predictions more transparent to human minds.

For example, in the early 1980s, interest arose in whether it might be possible to use quantum effects to signal faster than light, which according to Einstein's Theory of relativity would be impossible. The resolution to this problem turned out to hinge on whether it was possible to *clone* an unknown quantum state (Ref. [4]), i.e. construct a copy of a quantum state. If cloning was possible, then it would be possible to signal faster than light using quantum effects. However, cloning turned out to be impossible in general, in quantum mechanics.

Theorem 1 (No-cloning theorem). *In general, an exact copy of an arbitrary, unknown quantum mechanical state cannot be created.*

Proof. Assume we have an initial superposition of the quantum states $|\psi\rangle$ and $|\phi\rangle$, i.e. two arbitrary, unknown quantum states, $\frac{1}{\sqrt{2}}(|\psi\rangle + |\phi\rangle) \equiv |\psi\phi\rangle$.

Next assume there exists an unitary operator, U, such that $U|\psi\phi\rangle = |\psi\psi\rangle$. We may then construct the quantum state

$$|z\rangle = \frac{1}{\sqrt{2}}\left(|\psi_1\rangle + |\psi_2\rangle\right) \ , \tag{2.1}$$

where $|\psi_1\rangle$ and $|\psi_2\rangle$ are two, in general, different, arbitrary and unknown states. The effect of U on $|z\phi\rangle$ will then be

$$\begin{aligned} U|z\phi\rangle &= \frac{1}{\sqrt{2}}U\left(|\psi_1\phi\rangle + |\psi_2\phi\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|\psi_1\psi_1\rangle + |\psi_2\psi_2\rangle\right) \ . \end{aligned} \tag{2.2}$$

or

$$\begin{aligned} U|z\phi\rangle &= |zz\rangle \\ &= \frac{1}{2}\left(|\psi_1\psi_1\rangle + |\psi_1\psi_2\rangle + |\psi_2\psi_1\rangle + |\psi_2\psi_2\rangle\right) \ . \end{aligned} \tag{2.3}$$

Since Eq. (2.2) and Eq. (2.3), in general, are not equivalent, we have the contradiction that

$$U|z\phi\rangle \neq U|z\phi\rangle \ . \tag{2.4}$$

We therefore have to conclude that there exists no such operator, U, i.e. cloning is impossible. $\square$

A related contribution to the development of quantum computation and quantum information is the interest, that arose in the 1970s, in obtaining complete control over single quantum systems. Prior to the 1970s, applications of quantum mechanics typically involved control over a bulk sample containing an enormous amount of quantum mechanical systems, for instance gases or solids. Since the 1970s many techniques for controlling single quantum systems have been developed. Methods for trapping a single atom in an "atom trap", manipulating of single atoms with an electron microscope and electronic devices transferring only single electrons are examples of manipulations of single quantum systems.
The ability to control single quantum systems is essential if we are to exploit the power that lies within quantum computing and quantum information.

Despite the efforts to build quantum information processing systems, the results are modest. Experimental prototypes for doing quantum cryptography, a way of sharing secret information, have been demonstrated and are at the level where they may be put to use in daily life. However, it remains a great challenge for scientists and engineers of the future to develop large-scale quantum computers (Ref. [4]).

The modern incarnation of computer science was announced by the mathematician Alan Turing in a 1936 paper. Turing developed in detail an abstract notion of what we would now call a programmable computer, a model for computation known as the *Turing Machine*. Turing showed that there is a *Universal Turing Machine* that can be used to simulate any other Turing Machine. That is, if an algorithm can be performed on *any* piece of hardware or computational device, there exists an equivalent algorithm for a Universal Turing Machine, which performs exactly the same tasks as the original algorithm. This assertion, known as the *Church-Turing thesis*, asserts the equivalence between the physical concept of what class of algorithms can be performed on some physical device. with the rigorous mathematical concept of a Universal Turing Machine. The broad acceptance of this thesis laid the foundation for a rich theory of computer science.

Not long after Turing's paper, the first computers, developed from electronic devices, were introduced. Hardware development truly started in 1947, when the transistor was developed by John Bardeen, Walter Brattain and William Shockley (Ref. [4]). The power of computer hardware has ever since grown tremendously. *Moore's law*, stated by Gordon Moore in 1965, states that the computer power will double for constant cost, roughly every second year. Amazingly, Moore's law has held approximately true since the 1960s. However, already during the 1970s concern for the size of computers arose. The concern regarded the fact that computers grew ever smaller and soon would reach the limit for what was possible to produce. Twenty-thirty years later, whilst operating at nano-meter scale, we may laugh at the 1970s pessimists. However, we must still pay attention to the difficulties computer science soon will come by. The quantum mechanical tunnel effect and the Heisenberg's uncertainty relation will cause a lot of pain for digital circuit designers, when the distance between conductors become even smaller. Quantum mechanical effects may cause electrons to jump back and forth between two conductors, and in that way be the source of noise and corrupted signals.
Possible solutions to the problem posed by the failure of Moore's law is to move to a different computing paradigm. In the 1970s, multiple-valued logic was seen as such a paradigm (Ref. [5]). By leaving the realm of binary logic and Boolean algebra, one may process more information and still require less resources. In an electronic device consisting of eight conductors, one may represent the numbers 0 to $2^8 - 1 = 255$ with two voltage-levels. To repre-

sent approximately the same range of numbers with three voltage-levels, we just need five conductors. In the 1990s the quantum computer was expected to be the required change of paradigm. In this thesis we propose a paradigm based on both multiple-valued logic and quantum mechanical effects, namely a trinary quantum computer (see Section 2.2).

The idea of quantum computing is to use quantum mechanics to perform computations, instead of classical physics. It turns out that a classical, ordinary computer can be used to simulate a quantum computer, however it appears to be impossible to perform the simulation *efficiently*[1]. Thus quantum computers offer an essential speed advantage over classical computers. In a classical computer, a conductor may be either high or low (1 or 0, on or off), which means that they cannot do more than one computation at a time. Thus a classical computer is only capable of serial processing, i.e. if we are to perform a number of computations, the total time spent is equal to the sum of the individual computing times. One of the great assets of a quantum computer is that a quantum state may be both high and low at the same time, i.e. a superposition of for instance vacuum and a one-particle state ($\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$). This means that a quantum computer may derive many results at the same time, parallel processing. The total time spent equals the longest individual computing time. The speed advantage of the quantum computer is so great that many researchers believe that no conceivable amount of progress in classical computing would overcome the gap between the power of a classical computer and the power of a quantum computer (Ref. [4]).

As an example, let us consider the case where we want to perform the two binary-algebra computations $0 \cdot 1$ and $1 \cdot 1$ (see chapter 3). The classical computer will first compute $0 \cdot 1 \to 0$ and then $1 \cdot 1 \to 1$. Let us represent low, 0, by the vacuum state, $|0\rangle$, and high, 1, by the one-particle state $|1\rangle$. A quantum computer may now use the state $|input1\rangle = \frac{1}{\sqrt{2}}(|0\rangle \dot{+} |1\rangle)$ as one input and $|input2\rangle = |1\rangle$ as the other[2]. Assuming we have an operator U, such that[3] $U|xy\rangle = |x\ AND\ y\rangle|y\rangle$, where $|xy\rangle \equiv |x\rangle|y\rangle$, the quantum computer may perform the parallel process $U|input1\rangle|1\rangle = \frac{1}{\sqrt{2}}U(|01\rangle \dot{+} |11\rangle) \to \frac{1}{\sqrt{2}}(|0\rangle \dot{+} |1\rangle)|1\rangle$. Thus the quantum computer has performed both compu-

[1]Computations may be efficient or inefficient. Roughly speaking, an efficient algorithm is one which runs in time polynomial in the size of the problem solved. In contrast, an inefficient algorithm requires super-polynomial (typically exponential) time.

[2]$\dot{+}$ denotes ordinary addition of vectors, and $\cdot$ denotes logical AND.

[3]See Section 3.1.1 for definition of AND.

tations at the same time.

Another great asset of quantum computers is the ability to generate totally random numbers. If measuring the earlier described state, $|input1\rangle$, the result will, totally randomly, be either 1 or 0. For classical computers a number of random number-generators have been proposed, but they are all in principle deterministic. Whilst the classical computer calculates the "random" number (pseudo-randomness), and any other classical computer may predict which number will be chosen, a quantum computer picks random numbers 100% stochastically, and it cannot be predicted certainly what number will be chosen next. Obviously this is a big difference between a classical computer and a quantum computer, and a classical computer simulation of a quantum computer random number-generator will not be realistic. A number of algebraic algorithms uses randomness as an essential part of the algorithm. Such algorithms are of course impossible to perform on a deterministic, classical computer.

During the revolutionary development of computational devices and computational algorithms after Turing's paper in 1936, the Church-Turing thesis was modified at various occasions. The continuous ad-hoc adaption of the Church-Turing thesis motivated a doubt in whether it was possible to find a single model of computation which was guaranteed to be able to efficiently simulate any other model of computation. Motivated by this doubt, in 1985 David Deutsch asked whether the laws of physics could be used to derive an even stronger version of the Church-Turing thesis. Because the laws of physics are ultimately quantum mechanical, Deutsch was led to considering computing devices based upon the principles of quantum mechanics. These devices led ultimately to the modern concept of quantum computers. It is still not clear whether Deutsch's notion of a Universal Quantum Computer is sufficient to efficiently simulate an arbitrary physical system. This is one of the great, open problems for the field of quantum computing and quantum information theory. There are still unexplored realms of physics, e.g. string theory or quantum gravitation, that may take us beyond Deutsch's Universal Quantum Computer.

Deutsch was the first to prove that the powers of a quantum computer indeed exceeded the powers of classical computing devices. Deutsch's first step was subsequently followed by remarkable discoveries and algorithms. In 1994 Peter Shor demonstrated how a quantum computer efficiently could solve two extremely important problems, namely factorizing of large integers and the so-called "discrete logarithm" problem (Ref. [4]). In 1995 Lov Gover showed how search through some unstructured search space could also be sped up on a quantum computer. Both the results of Shor and Gover are strongly indicating that quantum computers are even more powerful than probabilistic Turing machines.

At about the same time as Shor's and Gover's algorithms were discovered, numerous physicists were following Richard Feynman's 1982 idea that quantum mechanical systems best would be simulated by quantum mechanical systems, or a computing device based upon quantum mechanical principles. Classical computers had revealed difficulties with simulating quantum mechanical systems. During the 1990s it was shown that it is indeed possible to use quantum computers to simulate quantum mechanical systems efficiently.

2.1.2 Quantum bits

The bit is the fundamental concept of classical computation and classical information. Quantum computation and quantum information are built upon an analogous concept, the quantum bit, or qubit for short. A classical bit may be in one out of the two states 0 and 1, which may be regarded as orthogonal states. Two possible orthogonal states for a qubit are the states $|0\rangle$ and $|1\rangle$, which may correspond to the classical states 0 and 1. The difference between bits and qubits is that a qubit can be in a state other than $|0\rangle$ or $|1\rangle$, namely a superposition of the two. In general, the qubit is of the form

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{2.5}$$

The numbers α and β are complex numbers and the special states $|0\rangle$ and $|1\rangle$ are known as computational basis states, and they form an orthogonal basis in a Hilbert space. A classical state may at all times be examined to determine whether it is 0 or 1. A qubit, however, cannot be examined in the same way. That is, we cannot determine α and β. When measuring a quantum state, we get either $|0\rangle$ or $|1\rangle$ with the probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. It turns out that only if infinitely many identical qubits were prepared and measured would one be able to determine α and β for a qubit in the state given in Eq. (2.5). This dichotomy between the unobservable state of a qubit and the observations we can make, lies at the heart of quantum computation and quantum information.

Despite the strangeness of qubits, they are verifiably real. Their existence are extensively validated by experiments, and many physical systems can be used to realize qubits. The two different polarizations of a photon, the alignment of nuclear spin in a magnetic field and two energy eigen-states of an electron orbiting a single atom, are some of the different systems that may be used.

Multiple qubits

Suppose we have two qubits. If these were classical bits, there would be four possible combinations, namely 00, 01, 10 and 11. So is the case with qubits. The computational basis is $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. Any quantum state may be expressed as a linear superposition of these,

$$|\phi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{2.6}$$

where we have the normalization condition

$$\sum_{i,j=0}^{1} |\alpha_{ij}|^2 = 1. \tag{2.7}$$

Similarly we may have n qubits, with 2^n possible combinations. The computational basis states of this system is of the form $|x_1 x_2 ... x_n\rangle$, where each x_i may be 0 or 1. This n-qubit state is to be compared to an n-bit bus in classical computers.

2.1.3 Quantum computation

Analogous to a classical computer, built from electrical circuits containing wires and logic gates, a quantum computer is built from quantum circuits containing wires and quantum gates. The quantum gates operate on the quantum states to alter the states and thereby manipulate the quantum information.
A variety of quantum gates have earlier been proposed, both for single qubit manipulations and for multiple qubit manipulations. However we will not study these here. The quantum gates studied in this thesis are designed without concern regarding previously proposed quantum gates.
There have been suggested quantum circuits to perform a variety of quantum algorithms, such as the factoring algorithm, searching algorithms and the quantum Fourier transform. These will also be left out of this thesis.

2.2 A New Way of Thinking?

2.2.1 Fundamental idea

The way of computing suggested in this thesis is not purely quantum mechanical, as are earlier quantum computers proposed. Some of the concepts involved may even be comprehended by considering classical physics only. The fundamental idea of this thesis is to propose a way of computing that holds the great advantages of quantum computing, yet is built upon many of the concepts of a classical computer. After all, classical computers have proven successful, and the methods and algebras used in classical computing have been tested and found working for more than half a century. The main concern of this thesis is therefore to create quantum states and quantum gates that make use of the successful methods of classical computing, yet achieves the powerful results of quantum computing.

In classical computation, the operations on states are, in general, not reversible. This is not possible in quantum computation. Quantum computation is achieved through unitary operations, and it can be shown that all unitary operators are invertible, i.e. all manipulations of quantum states are reversible. So is the case with the quantum gates proposed in this thesis, too. If we consider all outputs of the quantum gates described in this thesis, we may from the output states and the properties of the operators derive the input states. However, not all the outputs are interesting, and we will usually just throw away those we do not need, which results in irreversible operations.

2.2.2 Qubits or qutrits

As explained in the previous section, qubits are the quantum computational analogy to the bits of classical computation. In this thesis we will make use of a new concept, namely qutrits or quantum trits. Trits are the three possible states of a trinary algebra, and qutrits will be the quantum computational analogy.

In the youth of classical computing, it had still not been decided which number system to rely on. In Russia in the fifties, a trinary computer was built from magnetic cores wired in tandem (Ref. [9]). This computer, that was named Setun, after a nearby river, worked quite fine and was actually in use throughout the fifties and the sixties. Unfortunately, the technology was not yet capable of embracing the trinary concept. Using two cores, capable of three stable states, one could generate the numbers -1, 0 and 1 in trinary algebra (negative parallel, opposite and parallel magnetic fields), and the numbers 00, 01, 10 and 11 (0,1,2 and 3) in binary algebra. As we realize, the binary system was actually more efficient using electrical coils.

Over the years, the binary logic was preferred, as binary switching (Boolean) algebra is extremely well suited to mathematically describe electronic circuits.

As logic has been developed and electronic circuitry is approaching its phys-

ical limits, concerning size, alternatives to binary logic has been proposed. Theorists of logic state that a black and white (0s and 1s, trues and falses) formalism is not suited to describe the real world. That is, "maybe" ought to be a possible choice of answer. E.g. if we know that the sky is blue, and the grass is green, we cannot answer yes or no to the question "Is the car red?". We have to answer "maybe". Fuzzy logic and multiple-valued logic handles questions like this. In this thesis we will concentrate on a trinary logic, however, in principle, it is possible to expand into logic of higher order. The advantage of employing higher order logic is that each wire in a circuit may store more information. As a binary computer in an n-bit bus may represent the numbers 0 to $2^n - 1$, a trinary computer may represent the numbers 0 to $3^n - 1$. Such effects of a higher order logic would revolutionize both memory storing-capabilities, other storing devices and transferring of data over networks.
The disadvantages may be that trinary logic gates have proven both difficult and expensive to create. In a classical computer, a trinary logic could be implemented by introducing a negative voltage-level, witch requires the circuit to conduct in both directions, or by introducing another positive voltage-level. Both solutions have proven too complicated to realize commercially.

In this thesis we propose a solution based on coherent light (laser). The intensity of the laser-beam defines the magnitude of the trit. As shown in this thesis, the construction of logic gate-like quantum gates for coherent light is not so hard.

In any given number system, any number may be represented by the sum

$$... + d_3 r^3 + d_2 r^2 + d_1 r^1 + d_0 r^0, \tag{2.8}$$

where r is the base, or radix, and the coefficients d_i are the digits of the number representation. For instance does the decimal representation 65_{10}, where the subscript commonly is omitted, mean $6 \cdot 10^1 + 5 \cdot 10^0$. Writing the same number in a trinary base, we must write $2 \cdot 3^3 + 1 \cdot 3^2 + 0 \cdot 3^1 + 2 \cdot 3^0 = 2102_3$. As we see, the same number may be represented totally different in different computational bases. The number of digits used to represent a number in a given base is called the width and is denoted by the letter w. Obviously, a way of defining the cost of computation in a given base, is by calculating the product of radix, or available symbols (in trinary 3 and in hexadecimal 10), with the digits needed to represent a given number. That is, we want to minimize the expression

$$\text{cost} = r \cdot w \ , \tag{2.9}$$

when

$$
\begin{aligned}
r^w &= \text{constant} \\
&\Updownarrow \\
w &= \frac{\ln constant}{\ln r} \ . \qquad (2.10)
\end{aligned}
$$

Substitution of Eq. (2.9) in Eq. (2.10) gives

$$
\begin{aligned}
\text{cost} &= rw \\
&= r\frac{\ln \text{constant}}{\ln r} \ . \qquad (2.11)
\end{aligned}
$$

Treating r as a continuous variable, we may calculate

$$
\begin{aligned}
\frac{d \text{ cost}}{dr} &= 0 \\
\frac{d}{dr}\left(\ln \text{constant} \cdot \frac{r}{\ln r}\right) &= 0 \\
\text{i.e.} \quad r &= e \approx 2.718 \ . \qquad (2.12)
\end{aligned}
$$

So the number e minimizes the cost-function.
Since it is absurd to have a non-natural number as radix, we accept 3, which is closest to e, as the most effective base.

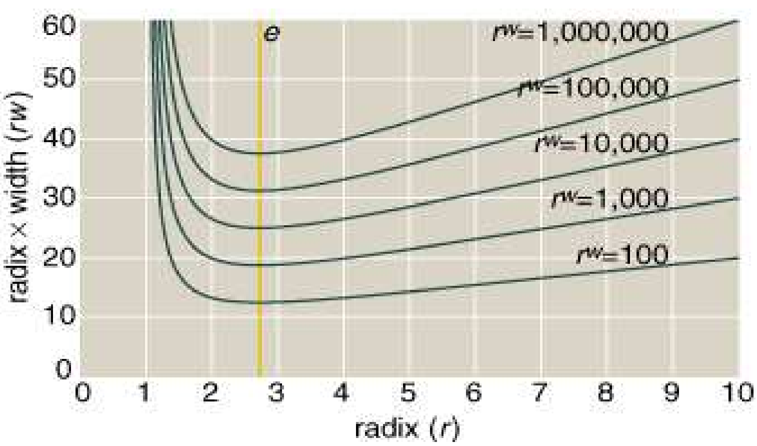


Figure 2.1: Most economical radix for a numbering system is e (about 2.718) when economy is measured as the product of the radix and the width, or number of digits, needed to express a given range of values. Here both the radix and the width are treated as continuous variables (Ref. [7]).

2.2.3 Qutrit representation by coherent states

As shown (see Section 1.2.3), coherent states with a large intensity of photons are near-orthonormal. Assuming the characteristic parameter large enough, we may suppress any cross-terms, and coherent states form an orthonormal set,

$$\langle \alpha | \beta \rangle = \delta_{\alpha,\beta}. \tag{2.13}$$

Provided α is a large number, we may represent our qutrits by the coherent states, $|0\rangle = vacuum$, $|1 \cdot \alpha\rangle$ and $|2 \cdot \alpha\rangle$. The three different states may easily be identified using a photon-detector.
Coherent states are incredibly resistant to noise. If an additional photon is added, or if a photon is annihilated, this does not alter the coherent state much. If one decided that the logic basis was to be vacuum, a one-photon state and a two-photon state, one cosmic photon could ruin the whole thing. In previous attempts of realizing quantum computers, decoherence has been a big issue. Suppose we want to use two-level atoms as our basic idea for computing. Let the excited state of the atom represent 1 and the ground state represent 0. Suppose we want to use a line of trapped atoms as a memory, storing 0's and 1's. We will quickly discover that our memory is no good. It is not stable enough, since the excited state of an atom is not stable, so the memory will quickly fade.
This is not a problem with the coherent states suggested in this thesis. Coherent states of light are extremely stable and resistant to noise/decoherence.

Analog computing

The ability to resist noise also enables the option of analog computing. A computer is analog when the physical representation of information, namely the intensity of a coherent state, it uses for computation is based on continuous degrees of freedom, instead of discrete logic values. For example, a thermometer is an analog computer. Analog computers have an infinite resource to draw upon in the ideal limit, since continuous variables may store an unlimited amount of information. In the presence of noise an analog computer is worthless. The presence of a finite amount of noise reduces the number of distinguishable states of a continuous variable to a finite number, and thus restricts analog computers to the representation of a finite amount of information. The possibility of optical analog computing, however, is not a subject of this thesis, and it is only mentioned as an opportunity.

2.2.4 Quantum parallelism

As mentioned, parallelism is one of the great advantages of quantum computing. We are not restricted to serial computation, as in classical computing, because we are not restricted to on-base logic states. As mentioned, a quantum state may be a superposition of, or in between, logic states, $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$. With the trinary, optical quantum computer, this advantage is stretched even further. A general quantum state may be expressed as a superposition of the three basis-states, $|\phi\rangle = \beta_0|0\rangle + \beta_1|1\cdot\alpha\rangle + \beta_2|2\cdot\alpha\rangle$. In addition to this advantage, the optical comprehension of a quantum computer makes use of the fact that coherent states of different "colour" are orthogonal. Since we have the operator relations Eq. (1.34) to Eq. (1.36), we can make use of either one of or a combination of the two following coherent quantum states

$$
\begin{aligned}
|\psi\rangle_{\text{n colours}} &= |\phi\rangle_{k_1} + |\phi\rangle_{k_2} + ... + |\phi\rangle_{k_n}, && (2.14)\\
|\psi\rangle_{\text{spin}} &= |\phi\rangle_{s_1} + |\phi\rangle_{s_2}, && (2.15)
\end{aligned}
$$

where k_i denotes the momentum of the ith state and s_1 and s_2 are two orthonormal directions of polarization. This means that we may mix two coherent states with either different polarization or different colour to create a quantum state that carries twice as much information as the single states alone. This is an opportunity that enables an incredible increase in the flow of data.

2.2.5 Quantum gates

This thesis does not at all look at the existing quantum gates, such as the Hadamard gate, the Fredkin gate (Ref. [4]) or any other as such. However, there is nothing principally wrong with the idea of extending the existing quantum logic into a base 3 (or even higher order) logic.
This thesis focuses on the development of gates that are similar to those used in classical computing, i.e. AND, OR and NOT.

Binary logic

As we will see in Section 3.2, there are algebraic properties that state

$$x \cdot 0 = 0 \cdot x = 0 \ , \tag{2.16}$$
$$x \cdot 1 = 1 \cdot x = x \ , \tag{2.17}$$
$$x + 0 = 0 + x = x \ , \tag{2.18}$$

and these properties define the binary algebraic product-type and sum-type operators. x may be any of the available logic values. For a two-valued algebra, with the available values 0 and 1, we may now construct the following two-input, binary, operators:

$a \cdot b$	0	1
0	0	0
1	0	1

and

$a + b$	0	1
0	0	1
1	1	?

.

As we can see, there is only one possible choice for a product-type operator, and this is the binary-logic AND-operator.
In the map of the sum-type operator, there is a choice. We may either choose $1 + 1 = 0$ or we may choose $1 + 1 = 1$. Both results are quite interesting and both operators are widely in use in binary logic.

$a + b$	0	1
0	0	1
1	1	1

is called the OR-gate, and

$a + b$	0	1
0	0	1
1	1	0

is called the XOR-gate.
All binary-logic gates may be constructed from either AND and OR or AND and XOR together with NOT.

Trinary logic

In Section 3.3 we will see that some of the most important properties of the trinary algebra are the relations

$$x \cdot 0 = 0 \cdot x = 0 \ , \tag{2.19}$$

$$x \cdot 2 = 2 \cdot x = x \ , \tag{2.20}$$

$$x + 0 = 0 + x = x \ . \tag{2.21}$$

These equations are in fact the definitions of *product-type* and *sum-type* operators (Ref. [8]) for a trinary logic. These constraints lead us to basic logic operations, $a \cdot b$ and $a + b$, that map as

$a \cdot b$	0	1	2
0	0	0	0
1	0	?	1
2	0	1	2

and

$a + b$	0	1	2
0	0	1	2
1	1	?	?
2	2	?	?

.

It is obvious that there are only three different product-type functions, namely those shown in Figure 2.2.

Since there are four open spaces in the sum-type map, there are $3^4 = 81$

(a)

$a \backslash b$	0	1	2
0	0	0	0
1	0	0	1
2	0	1	2

(b) MIN

$a \backslash b$	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

(c)

$a \backslash b$	0	1	2
0	0	0	0
1	0	2	1
2	0	1	2

Figure 2.2: Three-valued product-type functions.

different maps. However, the constraint $x + y = y + x$ reduces the number of maps to twenty-seven. If we demand that the functions fulfil both the commutative and associative laws as well, we reduce the total number to nine.

(a)

$a \backslash b$	0	1	2
0	0	1	2
1	1	0	2
2	2	2	2

(b)

$a \backslash b$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	0

(c)

$a \backslash b$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	1

(d)

$a \backslash b$	0	1	2
0	0	1	2
1	1	1	1
2	2	1	2

(e)

$a \backslash b$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	1

(f) MAX

$a \backslash b$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

(g) MODSUM

$a \backslash b$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

(h)

$a \backslash b$	0	1	2
0	0	1	2
1	1	2	1
2	2	1	2

(i) TSUM

$a \backslash b$	0	1	2
0	0	1	2
1	1	2	2
2	2	2	2

Figure 2.3: Three-valued sum-type functions.

All of these functions may be used as basic logic operations, but they are not all equally well suited. It can be shown that the product-type functions are equally good and that the MODSUM operation is the best suited sum-type operation (Ref. [8]), i.e. MODSUM in general, requires fewer product-terms (see Section 3.3), than the other sum-type functions to express an algebraic expression. The MAX-function is here actually, the least suited function.
It is however an important feature of the MAX-function that the binary-algebra OR-gate is included as a special case. The binary-algebra AND-gate is included in the MIN-function.

The gates provided by this thesis work on the basis states, $|0\rangle$, $|\alpha\rangle$ and $|2\alpha\rangle$, but not necessarily on superpositions of the basis states. Especially the gates that are intensity-dependent will experience problems. In some gates there has been assumed an existence of non-linear crystals with an intensity dependent transmittivity. In practice this means that the intensity, the number operator, $\hat{a}^\dagger\hat{a}$, is measured by the crystal. The crystal will be transparent to all states in the superposition whenever their total intensity surpass a given threshold-value, even though the gate was meant to transmit only some of the states.
This is a problem that will not be taken into account in this thesis. The gates have been shown to work on single states.
Solutions to the problem with superpositions may be to time-delay the different states of the superpositions, so that they don't appear at the crystal at the same time, to make superpositions of states with different frequency (see Eq. (2.14)) or of states with different polarization (see Eq. (2.15)).

Chapter 3

Introduction to Logic

3.1 Binary Logic

Binary logic (Ref. [3]) deals with variables that take on two discrete values and with operations that assume logical meaning. The two variables may be called by any names, but the most convenient, for mathematical purposes, are 0 and 1 (false and true). Binary logic may be used to describe, in a mathematical way, the processing of binary information and has been especially well suited for the analysis and design of digital systems. Binary logic is equivalent to the algebra called Boolean algebra.

3.1.1 Definition of binary logic

Binary logic consists of binary variables and logical operators. The variables are designated by letters and may take one out of two distinct possible values, 1 and 0. In binary logic there are three basic logical operations, AND, OR and NOT[1] AND and OR are so-called *binary operators*, i.e. operators that has two inputs. NOT is a *unary operator*, i.e. an operator that operates on one variable only.

1. AND is represented by the symbol $\cdot$ or by the absence of an operator, i.e. $x \cdot y \equiv xy \equiv x\ AND\ y$. The operator AND passes on the smallest of the two binary variables, i.e. if $z = x \cdot y$, $z = 1$ if and only if both x and y equals 1. Otherwise z equals 0. The AND operator may be represented by the map in Figure 3.1.

[1] Here OR is chosen as basic operator. As stated in Section 2.2.5, XOR would be an equally good choice.

AND	0	1
0	0	0
1	0	1

Figure 3.1: Map of the logical AND-operation.

2. OR is represented by the symbol $+$. OR passes on the larger of two variables, i.e. if $z = x + y$, $z = 0$ if and only if both x and y are equal to zero. Otherwise, z equals 1. OR may be represented by the map in Figure 3.2.

OR	0	1
0	0	1
1	1	1

Figure 3.2: Map of the logical OR-operation.

3. NOT is represented by a bar or a prime, i.e. $\bar{x} = x' = \text{NOT } x$. The operator passes on the "other possible" value, i.e. $\bar{0} = 1$ and $\bar{1} = 0$. We may represent NOT by the map in Figure 3.3.

X	$\bar{X}$
0	1
1	0

Figure 3.3: Map of the logical NOT-operation.

Binary logic resembles binary arithmetic. The symbols used for AND and OR are the same as those used for multiplication and addition. However, one must be careful not to confuse binary logic and binary arithmetic. An arithmetic variable designates a number that may consist of many digits. A logic variable is always either 1 or 0. For example, in binary arithmetic we have $1 + 1 = 10$, whereas in binary logic, we have $1 + 1 = 1$.

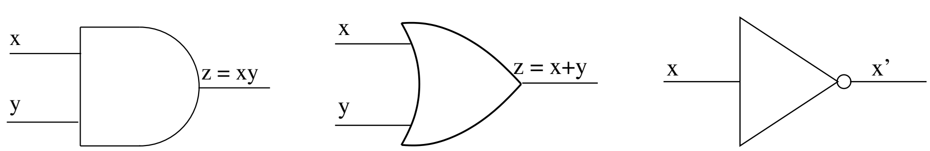
For each combination of the values of x and y, there is a value of z specified by the definition of the logical operation. These definitions may be listed in two-dimensional *maps* as shown in figures 3.1-3.3 or in *truth tables*, as shown in Table 3.1. A truth table is a table, of all possible combinations, of the possible values, of the input variables, showing the logical result of a given operation on the input variables. Truth tables corresponding to the maps in Figure 3.1-(3.3) would be the following tables

Table 3.1: Truth tables for logical AND, OR and NOT operations.

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	$\bar{x}$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

3.1.2 Logic gates

Electronic digital circuits are also known as *logic circuits*, because with the proper input they establish logic manipulation paths. Any desired information for computing or control can be operated upon by passing binary signals through various combinations of logic circuits. Blocks of hardware, containing electronic circuits, based on semi-conductor technology, that perform the logical operations of AND, OR and NOT have been invented and are widely in use in electronic digital design. The main concern of this thesis, however, is not electronics, and we will leave this topic unexplored. We will rather try to copy the properties of the logic operations. The logic circuits, called *gates*, are represented by the symbols shown in Figure 3.4, and we will adopt these symbols for later use.



(a) AND-gate (b) OR-gate (c) NOT-gate

Figure 3.4: Basic logic gates.

One may construct multiple-input logic gates from two or more basic gates as shown in Figure 3.5. The logic gate in Figure 3.5 will perform the AND

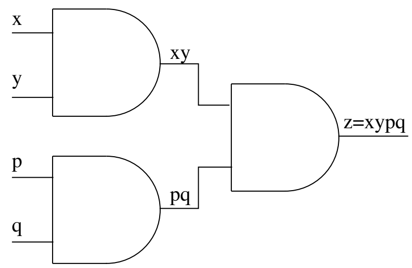


Figure 3.5: Multiple-input AND-gate construction.

operation on four inputs, i.e. the output, z, will be equal to the smallest of the four inputs.
For simplicity such blocks of logic gates are denoted with symbols similar to those in Figure 3.6. The mathematical system of binary logic is better known

(a) 4-input AND-gate

(b) 4-input OR-gate

Figure 3.6: Multiple-input logic gates.

as *Boolean algebra*, or switching algebra. This algebra is conveniently used to describe the operation of complex networks of digital circuits. Boolean algebra may be used to transform circuit diagrams into algebraic expressions and vice versa.

3.2 Boolean algebra

3.2.1 Axiomatic definition of Boolean algebra

In 1854 George Boole introduced a systematic treatment of logic (Ref. [3]) and developed for this purpose an algebraic system now called Boolean algebra. In 1938 C. E. Shannon introduced the two-valued Boolean algebra called *switching algebra* (Ref. [3]), in which he demonstrated that the properties of bistable electrical switching circuits can be represented by this algebra. The formal definition of Boolean algebra employs the Huntington postulates, formulated by E. V. Huntington in 1904.

Boolean algebra is an algebraic structure defined on a set of elements $\mathbf{B}$, together with two binary operators, $+$ and $\cdot$, provided the Huntington-postulates are satisfied:

1. (a) Closure with respect to the operator $+$.
   (b) Closure with respect to the operator $\cdot$.
2. (a) An identity element with respect to $+$, designated by 0: $x + 0 = 0 + x = x$.
   (b) An identity element with respect to $\cdot$, designated by $\mathbb{1}$: $x \cdot \mathbb{1} = \mathbb{1} \cdot x = x$.
3. (a) Commutative with respect to $+ : x + y = y + x$.
   (b) Commutative with respect to $\cdot : x \cdot y = y \cdot x$.
4. (a) $\cdot$ is distributive over $+ : x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
   (b) $+$ is distributive over $\cdot : x + (y \cdot z) = (x + y) \cdot (x + z)$.
5. For every element $x \in \mathbf{B}$, there exists an element $\bar{x} \in \mathbf{B}$ (called the complement of x) such that (a) $x + \bar{x} = \mathbb{1}$ and (b) $x \cdot \bar{x} = 0$.
6. There exists at least two elements x,$y \in \mathbf{B}$ such that $x \neq y$.

Comparing this algebra with ordinary algebra, we note the following differences:

1. The Huntington postulates do not include the associative law. However, this law holds for Boolean algebra and can be derived (for both operators) from the other postulates.
2. The distributive law of $+$ over $\cdot$, i.e. $x + (y \cdot z) = (x + y) \cdot (x + z)$, is valid for Boolean algebra, but not for ordinary algebra.
3. Boolean algebra does not have additive or multiplicative inverses; therefore, there are no subtraction or division operations.
4. Postulate 5 defines an operator called *complement* that is not available in ordinary algebra.

It can be seen from postulate 5 that a Boolean algebra must consist of an even number of elements.

Proof. First, assume that an element x is the complement of itself, i.e. $\bar{x} = x$. Theorem 1 states that we have $x + x = x \cdot x = x$ for any element x. From postulate 5 we then have $x + \bar{x} = x + x = x = 1$ and we also have $x \cdot \bar{x} = x \cdot x = x = 0$. So x is both 1 and 0, which is impossible.
An element cannot be the complement of itself.

Second, assume that two elements in **B**, $\bar{x}$ and x' have the same complement, x, i.e.

$$x + \bar{x} = 1 \wedge x \cdot \bar{x} = 0$$
$$x + x' = 1 \wedge x \cdot x' = 0$$

We then have the following

$$\begin{aligned}
\bar{x} &= \bar{x} + \bar{x} \\
&= \underbrace{\mathbb{1}}_{x+x'} \cdot \bar{x} + \bar{x} \\
&= \underbrace{x\bar{x}}_{0} + x'\bar{x} + \bar{x} \\
&= \bar{x} + \bar{x}x' + xx' \\
&= \bar{x} + \underbrace{(\bar{x} + x)}_{1} x' \\
&= \bar{x} + x'
\end{aligned}$$

This yields the same result for $\bar{x} \rightarrow x'$ so we get

$$\bar{x} = \bar{x} + x' = x'$$

If no element can be the complement of itself, and no element can be the complement of more than one element, the set **B** must consist of an even number elements. $\square$

3.2.2 Basic theorems and properties of Boolean algebra

The following theorems may be proven generally or by calculating every possibility, but neither will be done here (Ref. [3]).

Table 3.2: Postulates and Theorems of Boolean Algebra (note that $x' = \bar{x}$).

Postulate 2	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Postulate 5	(a) $x + \bar{x} = 1$	(b) $x \cdot \bar{x} = 0$
Theorem 1	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3, involution	$x'' = x$	
Postulate 3, commutative	(a) $x + y = y + x$	(b) $xy = yx$
Theorem 4, associative	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Postulate 4, distributive	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a) $(x + y)' = \bar{x}\bar{y}$	(b) $(xy)' = \bar{x} + \bar{y}$
Theorem 6, absorption	(a) $x + xy = x$	(b) $x(x + y) = x$

The operator precedence for evaluating Boolean expressions is (1) parentheses, (2) NOT, (3) AND, and (4) OR. This will later be adopted when we extend the algebra to the trinary case.
Any Boolean function may be represented in a truth table, and the truth table is unique. However, there may be numerous different algebraic expressions leading to the same truth table. As an example we may consider the binary algebra expressions

$$F_1 = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y} \ , \tag{3.1}$$

$$F_2 = x\bar{y} + \bar{x}z \ . \tag{3.2}$$

These expressions may at first sight seem different, but if we take a look at their truth tables (Table 3.3), we see that they are in fact equal.

Table 3.3: Truth table for the binary algebra-expressions $F_1 = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}$ and $F_2 = x\bar{y} + \bar{x}z$.

x	y	z	F_1	F_2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Because of the uniqueness of truth tables we deduce that the two algebraic expressions must be equal. This can be seen through some algebraic manipulations:

Table 3.4: Postulate/theorem applied to F_1 to obtain F_2.

Algebraic expression	Postulate or theorem applied
$F_1 = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}$	
$\bar{x}z\bar{y} + \bar{x}zy + x\bar{y}$	Postulate 3(b)
$\bar{x}z(\bar{y} + y) + x\bar{y}$	Postulate 4(a)
$\bar{x}z \cdot 1 + x\bar{y}$	Postulate 5(a)
$\bar{x}z + x\bar{y}$	Postulate 2(b)
$x\bar{y} + \bar{x}z = F_2$	Postulate 3(a)

In similar ways we may alter algebraic expressions to obtain different expressions with equal truth tables.

A binary variable may appear either in it's normal form (x) or in it's complement form ($\bar{x}$). Two binary variables combined with an AND operation may therefore appear in four different ways, (xy),($\bar{x}y$),($x\bar{y}$) or ($\bar{x}\bar{y}$), which each represents one distinct area of a Venn diagram (Ref. [3]).

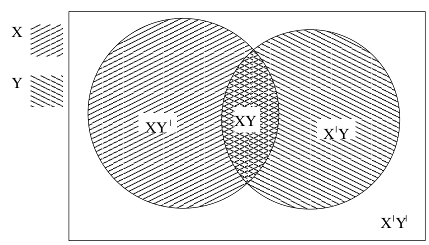


Figure 3.7: Example of a two-variable Venn diagram.

Each of the terms, (xy), ($\bar{x}y$), ($x\bar{y}$) and ($\bar{x}\bar{y}$), are referred to as a *minterm*, or a *standard product*. n variables may in a similar way be combined to form 2^n minterms.

Any Boolean function may be represented as a sum of minterms (sum meaning OR) (Ref. [3]).

2^n minterms may become a huge number of minterms, but one of the Huntington postulates saves us, namely postulate 5, which states $x + \bar{x} = 1$. This means that minterms that differ in one variable only may be simplified, i.e.

two minterms may be reduced to one minterm, $yx + y\bar{x} = y(x + \bar{x}) = y$. If this process is repeated, a complex algebraic form of minterms may be reduced to a quite simple one. This is the goal for the next section.

3.2.3 Simplification of Boolean functions

An important feature of Boolean algebra and it's relation to binary logic, is how an electronic circuit easily may be realized from an algebraic expression. Although the truth tables are unique, their algebraic equivalences may appear in many different forms, as shown in the previous section. To make efficient and economic electronic circuits it is important to minimize the algebraic expressions related to a given truth table, since the complexity of the algebraic expression is reflected in the complexity of the electronic circuit.
The map method provides a simple straightforward procedure for minimizing Boolean functions. The method was first proposed by Veitch and later modified by Karnaugh, and it is known as the "Veitch diagram" or the "Karnaugh map" (Ref. [3]).
The goal of the Karnaugh map is to express an algebraic expression as simply as possible in minterms. The method relies totally on postulate 5 and a quick description will be given below.

1. Express the algebraic form as a map. If there are more than two variables, groups of variables (ANDed together) may be regarded as single variables. Make sure that only one of the input variables of two neighbouring cells varies with 1, i.e. 01 and 11 may be neighbours, but 01 and 10 may not.

2. Encircle neigbouring 1's in the map. Note that only rectangular blocks of 1's may be encircled.

3. Each rectangular grouping of 1's now represents a minterm in the final, minimized algebraic expression. The rectangular shapes will contain minterms that only differ in one variable, and these may be simplified until they just contain variables that don't change.

Consider the following example:

Example of minimization

Let us consider the previous example of algebraic expressions. Let us minimize the expression $F_1 = \bar{x}\bar{y}z + \bar{x}yz + x\bar{y}$.

1. First we express the algebraic expression in a map:

		YZ			
		00	01	11	10
X	0	0	1	1	0
	1	1	1	0	0

Figure 3.8: Mapping of the truth-table in Table 3.3.

   Notice two things; first, there are numerous alternative ways to write down the map of F_1 by for instance changing the order of the variables, second that for two neigbouring cells, the expression yz never differs in more than one variable. This leads to the fact that the first and the last columns of the map are neighbours, as well.

2. We then encircle neigbouring 1's.

		YZ			
		00	01	11	10
X	0	0	1	1	0
	1	1	1	0	0

Figure 3.9: Encircling of neighbouring 1's.

   Notice that it is not necessary to encircle the minterms 001 and 101 since both are included in other groupings of minterms.

3. We write down the minimized expression[2]:

   $\boxed{\bar{x}\bar{y}z + \bar{x}yz} + \boxed{x\bar{y}\bar{z} + x\bar{y}z} = \bar{x}z + x\bar{y} = x\bar{y} + \bar{x}z,$

   which is exactly F_2. I.e. F_2 is the minimized expression for the truth-table given in Table 3.3.

The value of the minimization procedure is easily seen if comparing the electronic circuits resulting directly from the algebraic expressions (Figure 3.10 and Figure 3.11).

[2]In this expression $\bar{x}$ means $x = 0$ and x means $x = 1$, so that the expression $\bar{x}\bar{y}z$ means $0_x0_y1_z$, i.e. the cell where x=0, y=0 and z=1.

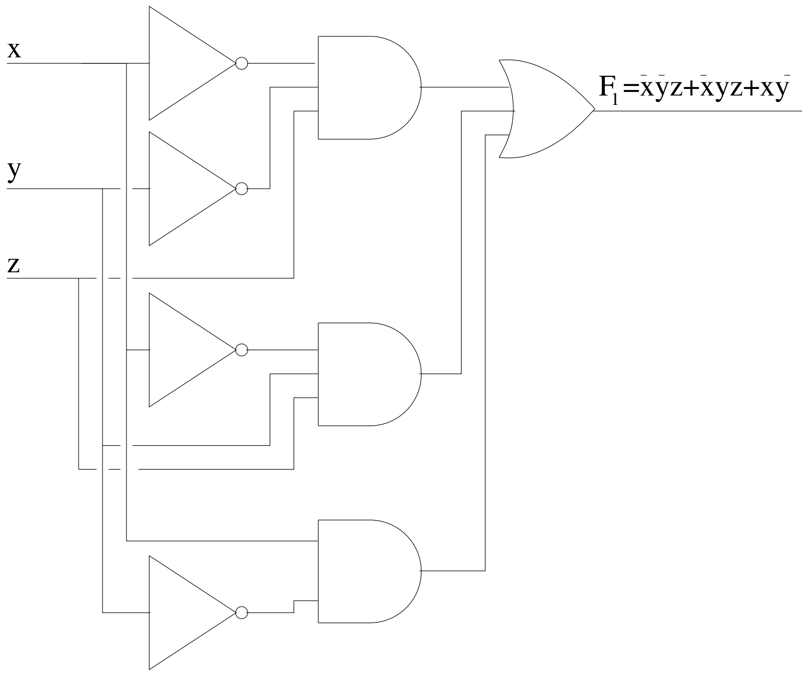


Figure 3.10: Electronic circuit resulting directly from the algebraic expression F_1.

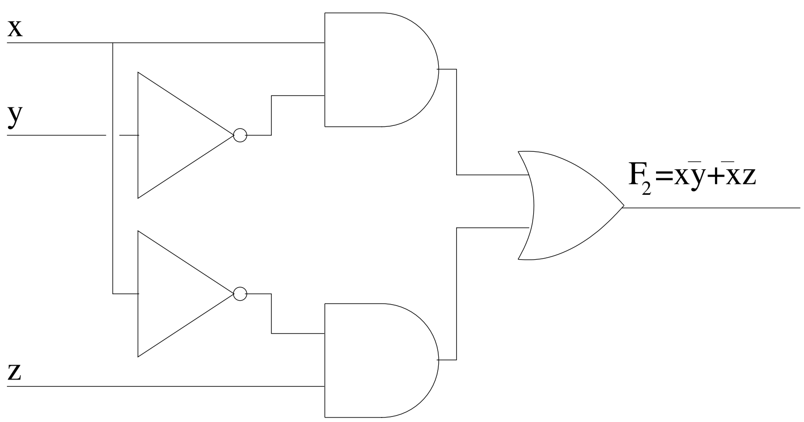


Figure 3.11: Electronic circuit resulting directly from the algebraic expression F_2.

3.3 Trinary Logic and Algebra

3.3.1 A trinary-valued switching algebra

The fundamental principle underlying multiple-valued logic systems is that they have n input variables and t output variables such that there are at most m values for each of the physical variables, where the physical values are $v_1, v_2, ..., v_m$ (Ref. [5]). Furthermore, if $v_1 < v_2 < ... < v_m$, the integer 0 can be assigned to v_1, 1 to v_2, ..., $m - 1 = p$ to v_m. Then each input and output variable may assume at any instant one of the set of m logic values from L, where $L = \{0, 1, 2, ..., p\}$ and $p = m - 1$. For a trinary logic system this leads to $L = \{0, 1, 2\}$ and $p = 2$.

As shown, the Boolean algebra cannot be used as an adequate model for the trinary-valued logic system, as it requires an even number of elements in L. The algebra used in this section is an implementable three-valued switching algebra.

We can now define two operations, $(+)$ and $(\cdot)$, by

Definition 1 (Max- and min-operations).

$$x + y = max(x, y)$$
$$x \cdot y = xy = min(x, y)$$

where $x, y \in L = \{0, 1, 2\}$.

From this definition it can easily be proven, either generally or by calculating all possibilities (see appendix A), that for any $x, y, z \in L$ the following properties are true:[3]

$Idempotent:$	$x + x = x,$	$xx = x;$
$Commutative:$	$x + y = y + x,$	$xy = yx;$
$Associative:$	$(x + y) + z = x + (y + z),$	$(xy)z = x(yz);$
$Absorbtion:$	$x + xy = x,$	$x(x + y) = x;$
$Distributive:$	$x + yz = (x + y)(x + z),$	$x(y + z) = xy + xz;$
$Nullelement:$	$x + 0 = x,$	$x0 = 0;$
$Universalelement:$	$x + 2 = 2,$	$x2 = x.$

As we can see, the only difference from the properties of a Boolean algebra is that of the unique complement. There are numerous possible unary[4] operations that can be defined, however only a few will be introduced here.

[3] We will always assume that $(\cdot)$ has precedence over $(+)$

[4] A unary operator is operating on one variable at a time. Accordingly, a binary operator has two inputs.

The unary operator (a, b) on the variable X, called a *literal* and denoted by $X(a, b)$, is given by

Definition 2 (Literal).

$$X(a,b) = \begin{cases} 0 & for \quad X \notin \{a,b\} \ , \\ 2 & for \quad X \in \{a,b\} \ . \end{cases}$$

The unary operator $^+$ on the variable X, called a *cycle* and denoted by X^+, is given by

Definition 3 (Cycle).

$$X^+ = X \dot{+} 1 \ mod 3 \ ,$$

where $\dot{+}$ *denotes arithmetic addition.*

The unary operator $^-$or $'$ on the variable X, called *complement* and denoted by $\hat{X}$ or X', is given by

Definition 4 (Complement).

X	$\hat{X}$
0	2
1	1
2	0

In addition to these binary and unary operations, we will in the Chapter 4 mention some other operations that may prove useful.

Before we move on, let us consider a simple example on the use of *literals*. Table 3.5 is formed in a manner completely analogous to the two-valued logic case. The left side of the table lists all possible values of $\alpha \in L^2$, i.e.. all possible combinations of two trinary logic variables. The right column lists the output values as a function of the two inputs.

Consider the $(X_1, X_2) = (1, 0)$ row of Table 3.5 and the product term $1 \cdot X_1(1,1) \cdot X_2(0,0)$. From the definitions of the *literal* and the *min* operators, $1 \cdot X_1(1,1) \cdot X_2(0,0)$ will take on the value 1 when $X_1 = 1$ and simultaneously $X_2 = 0$. For all other values of X_1 and X_2 the term will take the value 0, since either $X_1(1,1)$ or $X_2(0,0)$ will have the logic value 0. From this discussion it can be concluded that the $(X_1, X_2) = (1, 0)$ row is sufficiently represented by the term

$$1 \cdot X_1(1,1) \cdot X_2(0,0) \ .$$

Table 3.5: Table of combinations for a two-variable three-valued switching function.

X_1	X_2	$F(X_1, X_2)$
0	0	2
0	1	2
0	2	0
1	0	1
1	1	2
1	2	1
2	0	0
2	1	2
2	2	1

In general, any output function $F(X_1, X_2, ..., X_n)$ may be represented by the *max* (i.e. the logical sum) of product terms of the form

$$F(\alpha_t) \cdot X_1(i,i) \cdot X_2(j,j) \cdot ... \cdot X_n(k,k) \ , \tag{3.3}$$

where $i, j, ..., k$ are the values assigned to the variables of the function in the t^{th} row of the table of combinations; i.e., $\alpha_t = (i, j, ..., k)$. Only those terms not identically equal to zero need be included in the sum representing F. To illustrate the above, consider the function in Table 3.5. This function may be represented by the expression:

$$\begin{aligned} F(X_1, X_2) &= 2X_1(0,0)X_2(0,0) + 2X_1(0,0)X_2(0,1) + 1X_1(1,1)X_2(0,0) + \\ &\quad 2X_1(1,1)X_2(1,1) + 1X_1(1,1)X_2(2,2) + 2X_1(2,2)X_2(1,1) + \\ &\quad 1X_1(2,2)X_2(2,2) \\ &= 1\big(X_1(1,1)X_2(0,0) + X_1(1,1)X_2(2,2) + X_1(2,2)X_2(2,2)\big) + \\ &\quad 2\big(X_1(0,0)X_2(0,0) + X_1(0,0)X_2(1,1) + X_1(1,1)X_2(1,1) + \\ &\quad X_1(2,2)X_2(1,1)\big) \ . \end{aligned} \tag{3.4}$$

For the purposes of specifying an algorithmic minimization process it has been defined *implicants*, *prime implicants* and *subsuming*. One product term

$$r_1\Phi_1 = r_1X_1(a_1, b_1)X_2(a_2, b_2)...X_n(a_n, b_n) \ ,$$

subsumes a second product term

$$r_2\Phi_2 = r_2X_1(c_1, d_1)X_2(c_2, d_2)...X_n(c_n, d_n) \ ,$$

if and only if both
(1) $r_1 \leq r_2$ and (2) $c_i \leq a_i \leq b_i \leq d_i$ for all $X_i,\ i = 1, 2, ..., n$.

For simplicity, subsumes were defined with respect to two product terms in which all variables appear. Any product term not including some variable X_i may be replaced by another product term with $X_i(0, 2)$ included. It will be shown that in the final step of the minimization process, all literals of the form $X_i(0, 2)$ will be deleted. The definition of *subsumes* implies that $r_1\Phi_1(\alpha) \leq r_2\Phi_2(\alpha)$ for all $\alpha \in L^n$.
The product term $r\Phi$ is said to be an *implicant* of F if and only if
(1) $r\Phi(\alpha) > 0$ for some $\alpha \in L^n$ and (2) $r\Phi(\alpha) \leq F(\alpha)$ for all $\alpha \in L^n$.

An *implicant* is said to be *prime* if it *subsumes* no other implicant of F.

3.3.2 Map minimization

The map representation of a three-valued switching function is essentially just a rearrangement of it's table of combinations. An example of a map for a three-variable three-valued switching function is given in Figure 3.12
Note that if the function is not completely specified for a particular n-tuple input, the corresponding cell in the map contains a (-) and is called a *don't care-condition.* Let us take a look at the product term $r\Phi_r = r \cdot X_1(a_1, b_1)X_2(a_2, b_2)...X_n(a_n, b_n)$, which takes the value r for all input n-tuples, $\alpha(i, j, ..., k)$, such that $a_1 \leq i \leq b_1, a_2 \leq j \leq b_2, ..., a_n \leq k \leq b_n$. From the arrangement of the multiple-valued map it can be seen that these n-tuples for which $r \cdot \Phi_r = r$ constitute a rectangular grouping of cells in n-dimensional space. This rectangular grouping is of size $(b_1 - a_1 \dot{+} 1) \times (b_2 - a_2 \dot{+} 1) \times ... \times (b_n - a_n \dot{+} 1)$ cells[5].

The following algorithm for determination of just the prime implicants of a function directly from its map was proposed by *C. Michael Allen and Donald D. Givone* (Ref. [5]):

1. Set the logic value inside each cell containing a (-) to the logic value p (i.e. 2 in a three-valued algebra). Let k be an index starting with the value p.

2. Find all n-dimensional rectangular groupings of cells which have the logic value k, or higher, and which are not totally contained in any larger rectangular grouping of cells which have the logic value k, or

[5]The operation - denotes arithmetic subtraction, the operation $\dot{+}$ denotes arithmetic addition and the operation $\times$ denotes arithmetic multiplication

		X_2X_3		
		00	01	02
X_1	0	f(0,0,0)	f(0,0,1)	f(0,0,2)
	1	f(1,0,0)	f(1,0,1)	f(1,0,2)
	2	f(2,0,0)	f(2,0,1)	f(2,0,2)

		X_2X_3		
		10	11	12
X_1	0	f(0,1,0)	f(0,1,1)	f(0,1,2)
	1	f(1,1,0)	f(1,1,1)	f(1,1,2)
	2	f(2,1,0)	f(2,1,1)	f(2,1,2)

		X_2X_3		
		20	21	22
X_1	0	f(0,2,0)	f(0,2,1)	f(0,2,2)
	1	f(1,2,0)	f(1,2,1)	f(1,2,2)
	2	f(2,2,0)	f(2,2,1)	f(2,2,2)

Figure 3.12: Example of a map for three-variabled three-valued switching function.

higher. The product terms which correspond to these groupings are prime implicants if and only if they subsume no term previously found to be a prime implicant.

3. Set k to k-1.
4. If $k > 0$, repeat from step 2. Otherwise, just prime implicants have been found, so the process terminates.

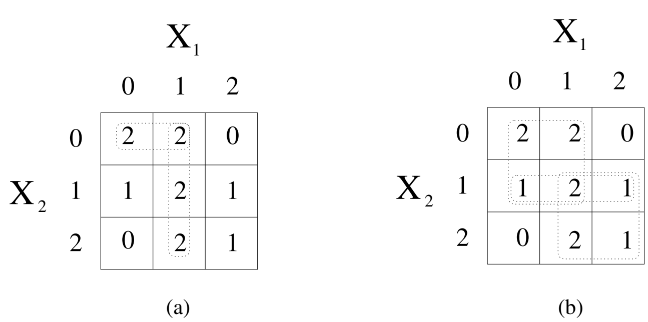


Figure 3.13: Prime implicants for the function of Table 3.5.

$$A : 2 \cdot X_1(0,0) \cdot X_2(0,1) \qquad C : 1 \cdot X_1(0,1) \cdot X_2(0,1),$$
$$B : 2 \cdot X_1(0,2) \cdot X_2(1,1) \qquad D : 1 \cdot X_1(1,1) \cdot X_2(0,2),$$
$$E : 1 \cdot X_1(1,2) \cdot X_2(1,2).$$

3.3.3 Example: Three-valued carry-free half-adder

A carry-free half-adder is a gate that carries out addition modulo B, where B is the base we are working in (2 in binary algebra, 3 in trinary algebra, etc.). Our base is 3, so the half-adder described here will carry out addition modulo 3. The truth table for such a gate is given in Table 3.6. The *carry*, that is left out here, since the gate is carry-free, takes care of whatever exceeds 2. For example, $2 + 2\ mod3 = 1$, so the output of this gate will in this case be 1, the carry would also have been one since $2 + 2 = 11$ in base 3. In other words, the carry takes care of “the next digit”.

Table 3.6: Truth table for three-valued carry-free half-adder.

X_1	X_2	$F(X_1, X_2) = X_1 \dot{+} X_2\ mod3$
0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	0
2	0	2
2	1	0
2	2	1

X_2 \ X_1	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Figure 3.14: Map of half-adder function in Table 3.6.

Let us now step by step follow the procedure described by *Allen and Givone* and construct a half-adder. Figure 3.14 shows the map for the function of Table 3.6. We note that there are no “don’t care-terms” in the map, so we may start at step 2 of the algorithm. First, all rectangular groupings of cells containing the value 2 are found. There are only three not-adjacent cells containing a 2, and these will each be counted as a rectangular grouping of one cell each. These are shown in Figure 3.15a along with their corresponding product terms. Next, all rectangular groupings of cells containing a 1 or a 2 are shown if Figure 3.15b along with their corresponding product terms.

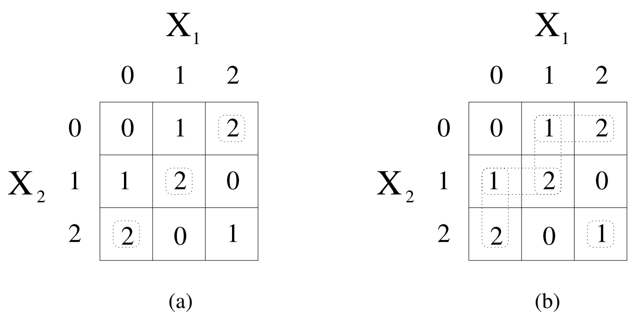


Figure 3.15: Prime implicants for the function of Table 3.6.

$$
\begin{array}{ll}
A : 2 \cdot X_1(2,2) \cdot X_2(0,0) & D : 1 \cdot X_1(1,2) \cdot X_2(0,0), \\
B : 2 \cdot X_1(1,1) \cdot X_2(1,1) & E : 1 \cdot X_1(0,1) \cdot X_2(1,1), \\
C : 2 \cdot X_1(0,0) \cdot X_2(2,2) & F : 1 \cdot X_1(0,0) \cdot X_2(1,2), \\
 & G : 1 \cdot X_1(1,1) \cdot X_2(0,1), \\
 & H : 1 \cdot X_1(2,2) \cdot X_2(2,2).
\end{array}
$$

Thus there are eight prime implicants of this function. It can be seen that all the prime implicants are needed for constructing this function. We may now write

$$
\begin{aligned}
F(X_1, X_2) &= A + B + C + D + E + F + G + H \\
&= 2 \cdot X_1(2,2) \cdot X_2(0,0) + 2 \cdot X_1(1,1) \cdot X_2(1,1) \\
&\quad +2 \cdot X_1(0,0) \cdot X_2(2,2) + 1 \cdot X_1(1,2) \cdot X_2(0,0) \\
&\quad +1 \cdot X_1(0,1) \cdot X_2(1,1) + 1 \cdot X_1(0,0) \cdot X_2(1,2) \\
&\quad +1 \cdot X_1(1,1) \cdot X_2(0,1) + 1 \cdot X_1(2,2) \cdot X_2(2,2) \\
&= 2 \cdot \big(X_1(2,2) \cdot X_2(0,0) + X_1(1,1) \cdot X_2(1,1) \\
&\quad +X_1(0,0) \cdot X_2(2,2) \big) \\
&\quad +1 \cdot \big(X_1(1,2) \cdot X_2(0,0) + X_1(0,1) \cdot X_2(1,1) \\
&\quad +X_1(0,0) \cdot X_2(1,2) + X_1(1,1) \cdot X_2(0,1) \\
&\quad +X_1(2,2) \cdot X_2(2,2) \big) \; . \qquad (3.5)
\end{aligned}
$$

And from this expression it is easy to construct the gate from the elementary gates literal, max and min. As seen in Figure 3.17, the construction becomes complicated, and we realize that techniques of simplification have to be found.

For binary logic the simplicity of the half-adder is stunning compared to the gate just described. The binary half-adder is shown in Figure 3.16.

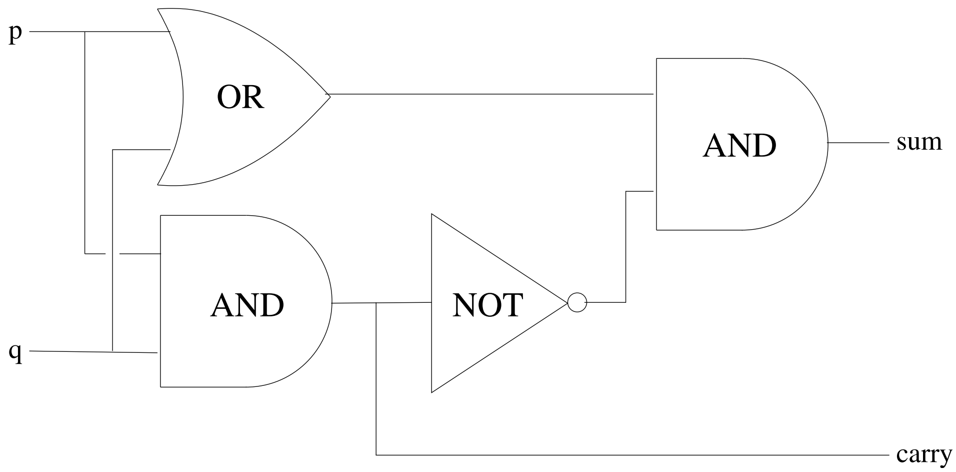


Figure 3.16: Scheme for constructing a base2 half-adder.

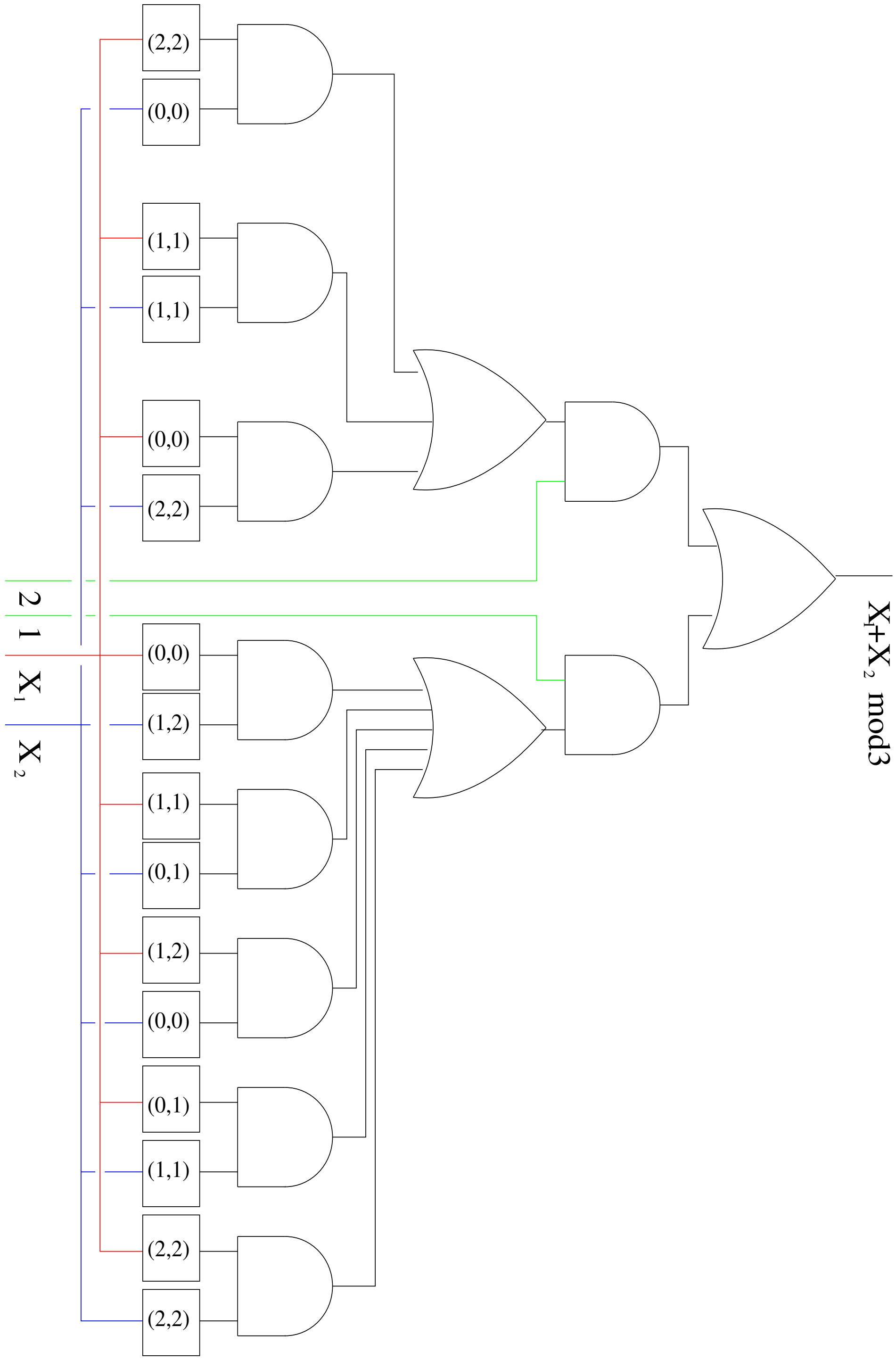


Figure 3.17: Scheme for constructing a base3 carry-free half-adder based on literal-, max- and min-gates.

Chapter 4

Quantum Optical Gates

If we take a look at a classical computer, it consists of small units, or gates. The most basic gates are AND, OR, XOR and NOT. In fact one may choose between OR and XOR, i.e. one may construct either one of the two from AND and the other of the two, together with NOT. The basic gates provide the operations given in the maps of Figure 4.1.

AND	0	1
0	0	0
1	0	1

(a) AND

OR	0	1
0	0	1
1	1	0

(b) OR

XOR	0	1
0	0	1
1	1	0

(c) XOR

p	NOT(p)
0	1
1	0

(d) NOT

Figure 4.1: Basic Boolean algebra operations.

It may easily be checked that

$$x\ XOR\ y = \bar{x}y + x\bar{y}. \tag{4.1}$$

If we take a look at these operations, we realize that they may be regarded in different ways. For instance, the AND-operation may either be regarded as a multiplication of the two inputs (MULT.), or it may be viewed as the operation giving the smallest of the two inputs (MIN). OR may be viewed as a gate putting out the largest of the two inputs (MAX), and XOR may be viewed as a sum modulo 2 (MODSUM). As was pointed out in Section 2.2.5, there are a variety of different product-type (AND) and sum-type (OR) operations when extending the algebra to higher orders. In this thesis, we provide the ideas necessary to construct the minimum-gate, the multiplication-gate, the maximum-gate and the MODSUM-gate for a three-valued algebra.

k\j	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

(a) MINIMUM

k\j	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

(b) MULTIPLICATION(mod3)

k\j	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

(c) MAXIMUM

k\j	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

(d) MODSUM

Figure 4.2: Some basic trinary-algebra operations.

4.1 Formalism

In this chapter, a series of schematic drawings are presented in an attempt to achieve the operations described in the previous section. In this section we will present the formalism and terminology used in the schematic drawings of Chapter 4. First of all, all incoming signals start at the left. All outgoing signals leave at the right of the drawing. Lines and arrows are representing the direction of movement of a coherent state (laser-beam), e.g. in an optical fiber.

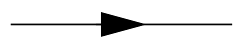


Figure 4.3: Representation of the direction of movement for a coherent state.

All input- and output-signals (states) are denoted by Dirac-formalism, ket-vectors, at their respective input-/output-ports.
The process

$$|output\rangle = |k'\alpha\rangle|j'\alpha\rangle = U|k\alpha\rangle|j\alpha\rangle \ , \tag{4.2}$$

will be denoted by the drawing in Figure 4.4, where the box with an U is an undefined circuit.

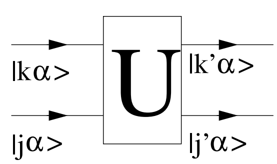


Figure 4.4: Representation of the process $|k\alpha\rangle|j\alpha\rangle \rightarrow U|k\alpha\rangle|j\alpha\rangle = |k'\alpha\rangle|j'\alpha\rangle$.

At several occasions, there will be output signals of no interest. Suppose there is a gate U, with two inputs, 1 and 2, and with three output-ports, a,b and c. This gate will perform the operation

$$|k\alpha\rangle_1|j\alpha\rangle_2 \to U|k\alpha\rangle_1|j\alpha\rangle_2 = |i\alpha\rangle_a|l\alpha\rangle_b|m\alpha\rangle_c \ , \tag{4.3}$$

where i,j,k,l and m are constants.
If the output at port c is of no interest, we will suppress the c-term and write

$$|i\alpha\rangle_a|l\alpha\rangle_b|m\alpha\rangle_c = |i\alpha\rangle_a|l\alpha\rangle_b \ . \tag{4.4}$$

Where a state is denoted by for instance $|k\alpha\rangle$, k is a variable that may take any of the values allowed (0,1 or 2 in a trinary base). Some of the gates may accept inputs other than those allowed in trinary algebra.
To show what the intensity of an "internal" state, a state within the circuit, the sloppy notation k, meaning $|k\alpha\rangle$ may be used.

The circuits described in this chapter are based on beam-splitters and non-linear crystals. The beam-splitters are balanced, i.e. $T = 1/2$ (see section (1.2.4). The beam-splitters will be denoted by the symbol in Figure 4.5.

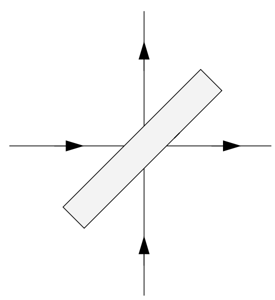


Figure 4.5: Representation of the 50/50 beam-splitter.

When we are interested in one of the two outputs only, we just neglect the other output as shown in Figure 4.6.
Non-linear crystals needed throughout this chapter may or may not exist, but there is nothing principally wrong in assuming their existence. In the case they do not exist, it might be possible to design gates that have the desired effects on coherent states. The theory of non-linear crystals will be left out of this text, since this thesis is meant as nothing more than a sketch of how optical computing may be carried out. Wherever non-linear crystals are used, the properties needed are specified. The non-linear crystals will be represented by the symbols in Figure 4.7. Some non-linear crystals need to be pumped, or "charged", by a laser in order to work. In some of the circuits

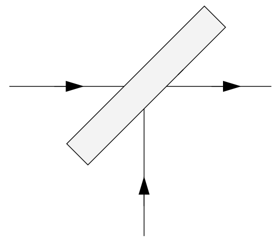
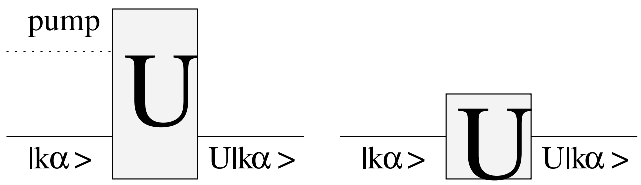


Figure 4.6: Representation of the 50/50 beam-splitter with one interesting output only.



(a) Pumped crystal (b) No pump

Figure 4.7: Representation of a non-linear crystal with the property U.

proposed, the pump, and thereby the properties of the crystal, is depending on the incoming coherent state. Effects we assume non-linear crystals may produce, are such as intensity dependent intensity-reduction and phase-shift and intensity independent intensity-reduction and phase-shift[1].

The symbol in Figure 4.8 will represent a 100% reflective mirror. We will assume that the mirrors do not alter any other property of the coherent states than their momentum.

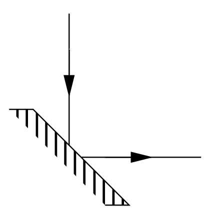


Figure 4.8: Representation of a 100% reflective mirror.

[1] A coherent state, $|k\alpha\rangle$, with intensity $k^2|\alpha|^2$ may for instance be altered to the state $|\frac{k}{\sqrt{2}}\alpha\rangle$ with the intensity $\frac{k^2}{2}|\alpha|^2$. Since α in general is a complex number, we may want to alter the phase of α, such that $\alpha \rightarrow \alpha \cdot e^{i\phi}$.

4.2 Specialized Gates

As we have seen, without minimization procedures, and maybe even with, circuits may become extremely complex, even for quite small operations. It would therefore be convenient to create gates that can perform more than the most elementary operations. This section will concentrate on such gates.

4.2.1 The linear light amplifier

In this section we shall concentrate on the simplest kind of amplifier, consisting of a partly inverted population of two-level atoms that effectively all see the same optical field (Ref. [1]). The population inversion is assumed to be maintained by some optical pumping scheme, so that it behaves as a reservoir.

Master equation for the amplifier field

We consider a system of N identical two-level atoms, of which N_2 are excited and N_1 are unexcited, interacting with a single mode quantum field. In order to make the problem as simple as possible, we shall suppose that we are dealing with an eigenmode of a free field, and that all atoms see essentially the same field. We assume that the field frequency is resonant with the atomic frequency, and we ignore the direct dipole interaction between atoms. We shall further assume that N_1 and N_2 are maintained approximately constant in time by some pump and loss mechanism. For the reduced density matrix, $\hat{\rho}$, it can be shown that the master equation in the interaction picture is

$$\frac{\partial \hat{\rho}}{\partial t} = -\frac{1}{2}A(\hat{a}\hat{a}^\dagger\hat{\rho} - \hat{a}^\dagger\hat{\rho}\hat{a} + h.c.) - \frac{1}{2}C(\hat{a}^\dagger\hat{a}\hat{\rho} - \hat{a}\hat{\rho}\hat{a}^\dagger + h.c.). \tag{4.5}$$

A represents the gain rate that is associated with the excited atom population, and C represents the loss rate that is associated with the unexcited atoms. For simplicity we shall take the gain and loss rates to be proportional to the populations N_2 and N_1, so that

$$\left.\begin{array}{l} A = 2\lambda N_2\ , \\ C = 2\lambda N_1\ , \end{array}\right\} \tag{4.6}$$

where λ is a rate that is of the order of the atomic line-width.

It can be shown (see Appendix C.1) that by making a diagonal coherent-state representation of the density operator $\hat{\rho}(t)$, namely

$$\hat{\rho}(t) = \int \phi(\nu, t)|\nu\rangle\langle\nu|d^2\nu, \tag{4.7}$$

substituting this representation into the master equation and replacing $\hat{a},\ \hat{a}^\dagger$ by differential operators, one can convert the operator equation for $\hat{\rho}$ into a c-number Fokker-Planck equation for $\phi(\nu,t)$. We may then look at the initial phase-space density $\phi(\nu,0)$ as representing the input field to the light amplifier whose output, after a time t, is $\phi(\nu,t)$. The equation now becomes

$$\frac{1}{\lambda}\frac{\partial\phi(\nu,t)}{\partial t} = -(N_2-N_1)\left[\frac{\partial}{\partial\nu}(\nu\phi(\nu,t)) + \frac{\partial}{\partial\nu^*}(\nu^*\phi(\nu,t))\right] + 2N_2\frac{\partial^2\phi(\nu,t)}{\partial\nu\partial\nu^*}. \tag{4.8}$$

We shall take this as the equation of motion, describing the linear light amplifier with input $\phi(\nu,0)$ and output $\phi(\nu,t)$.

The equation is analytically solvable, but we will merely present the answer in this text. It can be shown (see Appendix C.1) that the output field $\phi(\nu,t)$ is expressible as a simple convolution of the input, $\phi(\nu,0)$, with the phase-space density $\phi_s(\nu,t)$ of a thermal field that is associated with spontaneous emission. Thus

$$\phi(\nu,t) = \int \phi_0(\nu')\phi_s(\nu - G(t)\nu',t)d^2\nu', \tag{4.9}$$

where

$$\phi_s(\nu,t) \equiv \frac{1}{\pi m(t)}e^{-|\nu|^2/m(t)}\ , \tag{4.10}$$

$$m(t) \equiv \left(\frac{N_2}{N_2-N_1}\right)\left[|G(t)|^2-1\right]\ , \tag{4.11}$$

$$G(t) \equiv e^{(N_2-N_1)\lambda t}e^{-i\omega t}. \tag{4.12}$$

It is evident that $m(t)$ represents the average photon number of the thermal or spontaneous emission field. It will be shown that $G(t)$ represents the amplifier gain after a time t.

If we make the change of variable $\nu - G(t)\nu' = \nu''$ in Eq. (4.9) and write

$$\phi(\nu,t) = \int \phi_0\left(\frac{\nu-\nu''}{G}\right)\phi_s(\nu'')d^2\nu'', \tag{4.13}$$

then it is apparent from the convolution structure that the output of the light amplifier can be regarded as resulting from the interference of the amplified input field with the spontaneous emission field. For example if we calculate the average output field amplitude $\langle \hat{a}(t) \rangle$ at time t from Eq. (4.9), we have

$$\begin{aligned}
\langle \hat{a}(t) \rangle = \langle \nu \rangle_t &= \int \nu \phi(\nu, t) d^2\nu \\
&= \int \int (\nu - G(t)\nu')\phi_s(\nu - G(t)\nu', t)\phi_0(\nu') d^2\nu' d^2\nu \\
&+ G(t) \int \int \nu' \phi_0(\nu')\phi_s(\nu - G(t)\nu', t) d^2\nu' d^2\nu.
\end{aligned}$$

The first integral on the right yields zero because the average value of the thermal field is zero (Eq. (C.8)), i.e.

$$\int \nu \phi_s(\nu) d^2\nu = 0,$$

whereas the second term gives

$$G(t) \int \nu' \phi_0(\nu') d^2\nu' = G(t)\langle \nu' \rangle_0,$$

because the integral over ν yields unity. Finally

$$\langle \hat{a}(t) \rangle = \langle \nu \rangle_t = G(t)\langle \hat{a}(0) \rangle. \tag{4.14}$$

Hence the average output field equals the average input field multiplied by $G(t)$, which is the amplifier gain, because the spontaneous emission field is zero.

The spontaneous emission field does however contribute to the average photon number $\langle \hat{n}(t) \rangle$ at the output. Thus we have with Eq. (4.9) with the help of the optical equivalence theorem (Ref. [1]),

$$\begin{aligned}
\langle \hat{n}(t) \rangle = \langle \hat{a}^\dagger(t)\hat{a}(t) \rangle &= \langle |\nu|^2 \rangle_t \\
&= \int |\nu|^2 \phi(\nu, t) d^2\nu \\
&= \int \int |\nu|^2 \phi_s(\nu, t)\phi_0(\nu') d^2\nu' d^2\nu \\
&+ |G(t)|^2 \int \int |\nu'|^2 \phi_0(\nu')\phi_s(\nu, t) d^2\nu' d^2\nu \\
&+ G(t) \int \int \nu^* \nu' \phi_0(\nu')\phi_s(\nu, t) d^2\nu' d^2\nu \ + \ c.c. \\
&= m(t) \ + \ |G(t)|^2 \langle \hat{n}(0) \rangle,
\end{aligned} \tag{4.15}$$

because the two last integrals on the right vanish. Thus the amplification contributes $|G(t)|^2\langle \hat{n}(0) \rangle$ photons and the spontaneous emission contributes m(t) photons to the output field on the average.

Input-output correlations

In order to show that, except for the spontaneous emission, the amplifier output field is coherent with the input field, we now calculate the input-output cross-correlation function $\langle \hat{a}^\dagger(0)\hat{a}(t)\rangle$. For this purpose we need to construct the Green function $\mathcal{G}(\nu, t|\nu_0, 0)$, or the conditional phase space density, from Eq. (4.9) by putting $\phi_0(\nu') = \delta^2(\nu' - \nu_0)$. We then obtain the equation

$$\mathcal{G}(\nu, t|\nu_0, 0) = \phi_s(\nu - G(t)\nu_0, t) = \frac{1}{\pi m(t)} e^{-|\nu - G(t)\nu_0|^2/m^2(t)}, \tag{4.16}$$

and it follows that

$$\begin{aligned} \langle \hat{a}^\dagger(0)\hat{a}(t)\rangle &= \int\int \nu_0^* \nu \mathcal{G}(\nu, t|\nu_0, 0)\phi_0(\nu_0) d^2\nu_0 d^2\nu \\ &= \int\int \nu_0^*(\nu' + G(t)\nu_0) \frac{1}{\pi m(t)} e^{-|\nu - G(t)\nu_0|^2/m^2(t)} d^2\nu_0 d^2\nu', \end{aligned}$$

where we have used Eq. (4.16) and have put $\nu = \nu' + G(t)\nu_0$. Hence

$$\left.\begin{aligned} \langle \hat{a}^\dagger(0)\hat{a}(t)\rangle &= G(t)\langle \hat{n}(0)\rangle, \\ \text{or} \qquad \langle \nu^*(0)\nu(t)\rangle &= G(t)\langle \nu^*(0)\nu(0)\rangle, \end{aligned}\right\} \tag{4.17}$$

because the integral involving the term $\nu_0^*\nu'$ vanishes. The input-output cross-correlation therefore coincides with the input autocorrelation except for the amplifier gain factor $G(t)$. Hence the output state is coherent for a coherent input state.

Conclusion

The linear amplifier will provide important features in an optical quantum computer. As the quantum states evolve in quantum gates one may experience loss of intensity. Due to the intensity sensitivity of some of the gates, problems may occur. With the linear amplifier however, the original intensity of the laser beam may be restored. An important matter is the frequency sensitivity of the amplifier. One of the great assets of a quantum computer is the ability of parallel processing, and one way to conduct this is to mix states of different colour, i.e. different frequency. To make use of this asset, we will be needing a linear amplifier that can deal with all the necessary frequencies. We will during this paper assume that this is only a matter of technology, and we will assume that such an amplifier may be constructed.

We will represent a linear amplifier with gain factor G, by the symbol in Figure 4.9.

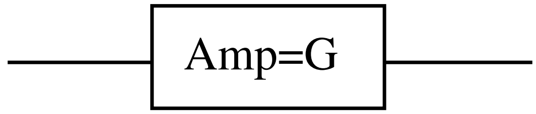


Figure 4.9: Symbol representing a linear amplifier with gain factor G.

4.2.2 Addition and subtraction

Arithmetic addition of two numbers is easily achieved in the proposed paradigm. As previously mentioned, the coherent states make an excellent basis for analog computing (see Section(2.2.3)). Arithmetic addition of both analogous signals and qutrits may prove useful, and may be achieved by the same gate.

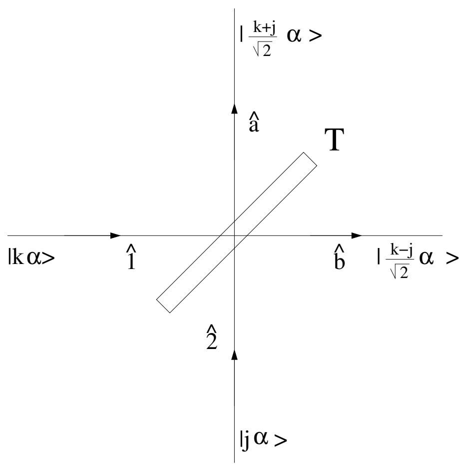


Figure 4.10: Schematic drawing of an arithmetic adder/subtractor.

The addition operation, as well as subtraction, may be achieved by making use of a 50/50 beam-splitter (and an amplifier). As described in Section (1.2.4), a beam-splitter has two inputs and two outputs. The outputs may be understood as weighted addition and subtraction of the two inputs respectively. When employing a 50/50 beam-splitter, $T = 1/2$, we achieve balanced addition/subtraction such that

$$|input\rangle = |k\alpha\rangle_1 |j\alpha\rangle_2 \rightarrow |output\rangle = |\frac{k+j}{\sqrt{2}}\alpha\rangle_a |\frac{k-j}{\sqrt{2}}\alpha\rangle_b \ . \tag{4.18}$$

We realize that k and j may be continuous, complex variables, thus being a base for analogous, complex computing, or they may be discrete variables, thus being a base for discrete computation. The output of the beam-splitter is of course not restricted to any base of computation, other than that of the complex number system. We therefore have to employ other gates as well to restrict the output to the base we are working in. The modulo 3-gate, described in Section (4.2.4), and the linear amplifier, described in Section (4.2.1), are useful gates in this case. If working with analogous computing, there is no obvious need to modify the output.

Addition modulo 3 will, in a trinary algebra, result in the map of Figure 4.11.

$k\backslash j$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

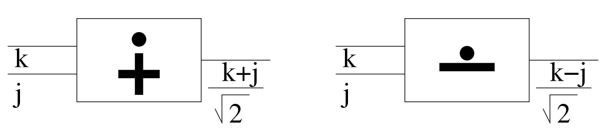
Figure 4.11: Map of addition modulo 3.

If analyzing the binary logic XOR-gate,

$x\backslash y$	0	1
0	0	1
1	1	0

, we notice that this gate actually performs addition modulo 2. Thus addition modulo 3 may be viewed as a trinary algebra equivalent to the XOR-gate. An important feature, however, of the basic gates proposed in Section 4.3 is that the binary operators are included in the trinary operators. This is as we easily realize not the case with XOR and addition modulo 3, since 1 added to 1 modulo 3 equals 2 and 1 XOR 1 equals 0.

The addition-gate and the subtraction-gate will be denoted by the symbols in Figure 4.12



(a) Adder (b) Subtractor

Figure 4.12: Symbols for arithmetic addition and subtraction.

4.2.3 Square-gate

As has been pointed out, it will prove convenient to construct specialized gates, since these may require less resources than the equivalent circuits designed from the basic binary operators, AND and OR. This section will provide two unary operations that may prove useful.

Definition 5 (Larger than/less than). *There exist two binary relations referred to as* larger than *and* less than *and denoted by* $>$ *and* $<$ *respectively, that apply to trinary algebra.*
The expression $x > y$ *is read as* x larger than y *and the expression* $x < y$ *is read as* x less than y.
The relations are to be considered as inverses of each other such that if $x < y$ *holds,* $y > x$ *is also true.*

Definition 6 (Magnitude of trinary numbers). *Based on the previous definition of relations, we now define the* order *of the elements of the trinary algebra as*

$$0 < 1 < 2\ . \tag{4.19}$$

Definition 7 (Linear unary operator). *Assume there exists a unary operator,* $\mathbb{O}$ *with the property* $\mathbb{O}i_x = o_x$.
The operator $\mathbb{O}$ *is called a* linear unary operator *if and only if*

$$\begin{array}{ccc} i_x & < & i_y \\ & \Updownarrow & \\ o_x & < & o_y \end{array}\ , \tag{4.20}$$

for all possible inputs.
If this is not true, the operator $\mathbb{O}$ *is called a* non-linear unary operator.

The beam-splitter is a linear optical medium, and hence we cannot construct a non-linear unary operator from beam-splitters alone. However beam-splitters and non-linear crystals may together achieve non-linear effects.
In the construction below, we use a non-linear medium that gives a phase shift depending on the intensity of the control- or *pump*-trit. As a special case we leave the trinary domain, and the control-trit may be either -1,0 or 1. Such a control trit is easily created from a normal trinary input by subtracting 1 (see Section 4.2.2). The phase shift, caused by the non-linear crystal, will be $i\pi$ plus an additional $i\pi c$, where $c \in \{-1, 0, 1\}$. As we easily realize, control-trit $c = 0$ is the only case where we experience an actual

phase shift. If we represent the three possible inputs in a vector, we may now easily create a gate that performs the non-linear operation

$$\mathbb{O}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} . \tag{4.21}$$

We realize that the operator $\mathbb{O}$ squares the input, which is a non-linear operation. Obviously, the order[2] of the outputs is not the same as the order of the corresponding inputs. This non-linear unary operation may act as a source for non-linear effects. The output may easily be manipulated to suit different needs. As we shall see we may easily expand the circuit to square trinary numbers.

The operator may be realized as the gate in Figure 4.13

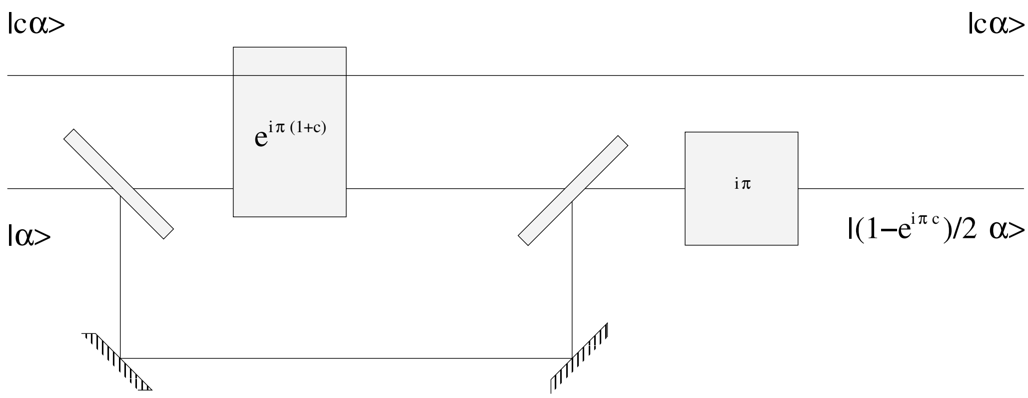


Figure 4.13: Gate that squares the inputs (-1 0 1).

The gate in Figure 4.13 produces the truth-table

c	$\frac{1-exp(i\pi c)}{2}$
-1	1
0	0
1	1

To square the elements of the trinary algebra, we have to manipulate quite a bit with the previously described gate [3].

[2] Order as defined in definition 6, but extended to involve -1 as well, $-1 < 0$

[3] The input of a gate may be one and only one out of three different values, i_0, i_1 and i_2, and these are represented in a row-vector, $i_0 \;\; i_1 \;\; i_2$. When operating on the input value, the operation is described at the left of the table, and the resulting output is written down as a row vector also, $o_0 \;\; o_1 \;\; o_2$, where the indices relate.

First add the two outputs of the previously described gate.		1	0	1
	$\dot{+}$	-1	0	1
Then add the ordinary trit corresponding to the squared input, (0 1 2).		0	0	2
	$\dot{+}$	0	1	2
And we get the square of (0 1 2)		0	1	4

To square a trinary number (0,1 or 2) one may therefore construct the following circuit. Notice that the output is not a trinary number and may need to be modified by the modulo3-gate described in Section 4.2.4.

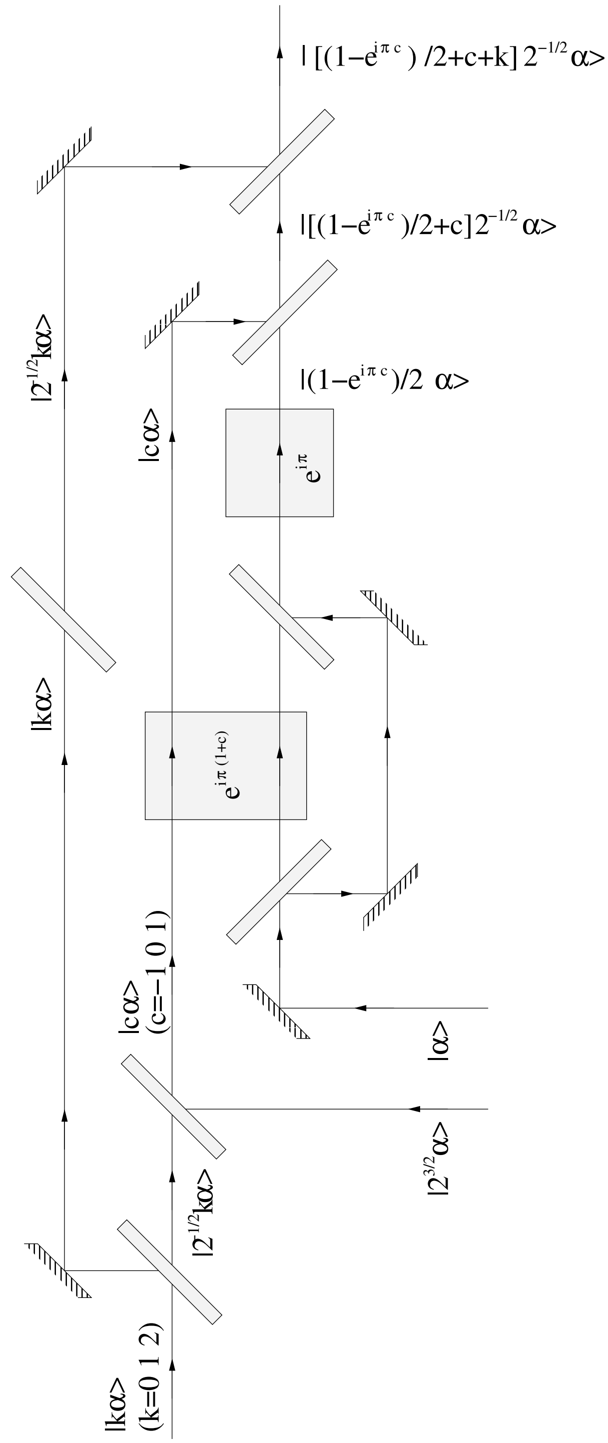


Figure 4.14: Gate that squares k=(0 1 2).

4.2.4 Modulo 3-gate

One of the possibilities we have when using "optical electronics", is that we may add any two numbers, i.e. we are not restricted to a number system, and both the input and the output may be arbitrarily large. In classical electronics, calculations are restricted to the binary number system. Neither inputs nor outputs of gates may exceed 1. In the quantum optical gates proposed in this thesis we are not principally restricted to any number system. In an adder (Section 4.2.2) there are no problems associated with adding numbers that are not contained in the binary, trinary or any other limited system. The answer may be "outside" the system we are operating in as well. However, in this thesis we have decided to explore the realms of trinary logic and trinary algebra. Analogously to the classical binary computers ($1 \dotplus 1 = 2\ mod2 = 0$),[4] we wish to stay within the trinary system, such that all calculations will be modulo 3 ($1 \dotplus 2 = 3\ mod3 = 0$ and $2 \dotplus 2 = 4\ mod3 = 1$).

In a binary computer any number $X = a_8 2^8 + a_7 2^7 + ... + a_1 2^1 + a_0 2^0$ may be represented as the number $a_8 a_7 ... a_1 a_0$ in an 8-bit bus (a system of 8 separate conductors each holding a voltage of 0 or 1).
We want to copy the $mod2$ properties of the classical computers and it would therefore be convenient to design a gate that makes sure we stay within the boundaries of trinary algebra. This gate will be known as the *modulo 3-gate*, and the operation will be denoted by $mod\ 3$, i.e. $X\ mod\ 3$ means that the modulo 3-operation is performed on the number X. Assuming we take care and never exceed the natural number 5, it is sufficient that the modulo 3-gate has the arithmetic properties

$$X\ mod\ 3 \simeq const \cdot \begin{cases} X & 0 \leq X < 3\ , \\ X - 3 & 3 \leq X < 6\ , \end{cases} \tag{4.22}$$

where *const* is an arbitrary constant.
We must stress the fact that the modulo 3-gate is not designed to cope with inputs larger than 5. Because of the nature of the non-linear media used, we get into trouble if trying to operate on larger numbers. One must be sure to apply the modulo3-gate wherever and whenever the output intensity of a gate or circuit may exceed the equivalent intensity of the number 2.
By combining the modulo 3-gate and adder-gates, we may for instance build half- and full adders that are similar to those used in classical binary computers.

[4] $\dotplus$ means arithmetic addition

In constructing the following modulo 3-gate we have assumed the existence of a non-linear optical media which modifies the phase of the input signal, depending on the intensity of a *pump*-laser. The wanted effect may possibly be achieved by using a *Kerr-medium*. The Kerr effect is a nonlinear process involving the third-order nonlinear polarisability of a nonlinear medium (Ref. [2]). The field undergoes an intensity dependent phase shift.
The following construction requires a medium that gives a phase shift of π whenever the medium is "turned off", m,n=0. When the medium is "turned on", m,n=1, the phase shift is zero.
We will also need a medium that simply gives a phase shift of π and a medium that has a constant transmittivity of less than 1.

m and n are given by

$$m = \begin{cases} 1 & for\ \frac{k}{6} < 2\ , \\ 0 & for\ \frac{k}{6} \geq 2\ , \end{cases} \tag{4.23}$$

and

$$n = \begin{cases} 1 & for\ k-3 < 0\ , \\ 0 & for\ k-3 \geq 0\ . \end{cases} \tag{4.24}$$

Furthermore are

$$f(k) = \begin{cases} 0 & for\ k < 3\ , \\ \frac{3-k}{2} & for\ k \geq 3\ , \end{cases} \tag{4.25}$$

$$g(k) = \begin{cases} \frac{\sqrt{2}\ k}{6} & for\ k < 3\ , \\ 0 & for\ k \geq 3\ , \end{cases} \tag{4.26}$$

and the output is given by

$$h(k) = \frac{g(k) - \frac{\sqrt{2}}{3} f(k)}{\sqrt{2}} = \begin{cases} \frac{k}{6} & for\ k < 3 \\ \frac{k-3}{6} & for\ k \geq 3 \end{cases} \tag{4.27}$$

$$= {}^{1}\!/\!{}_{6} \cdot (k\ modulo\ 3)\ .$$

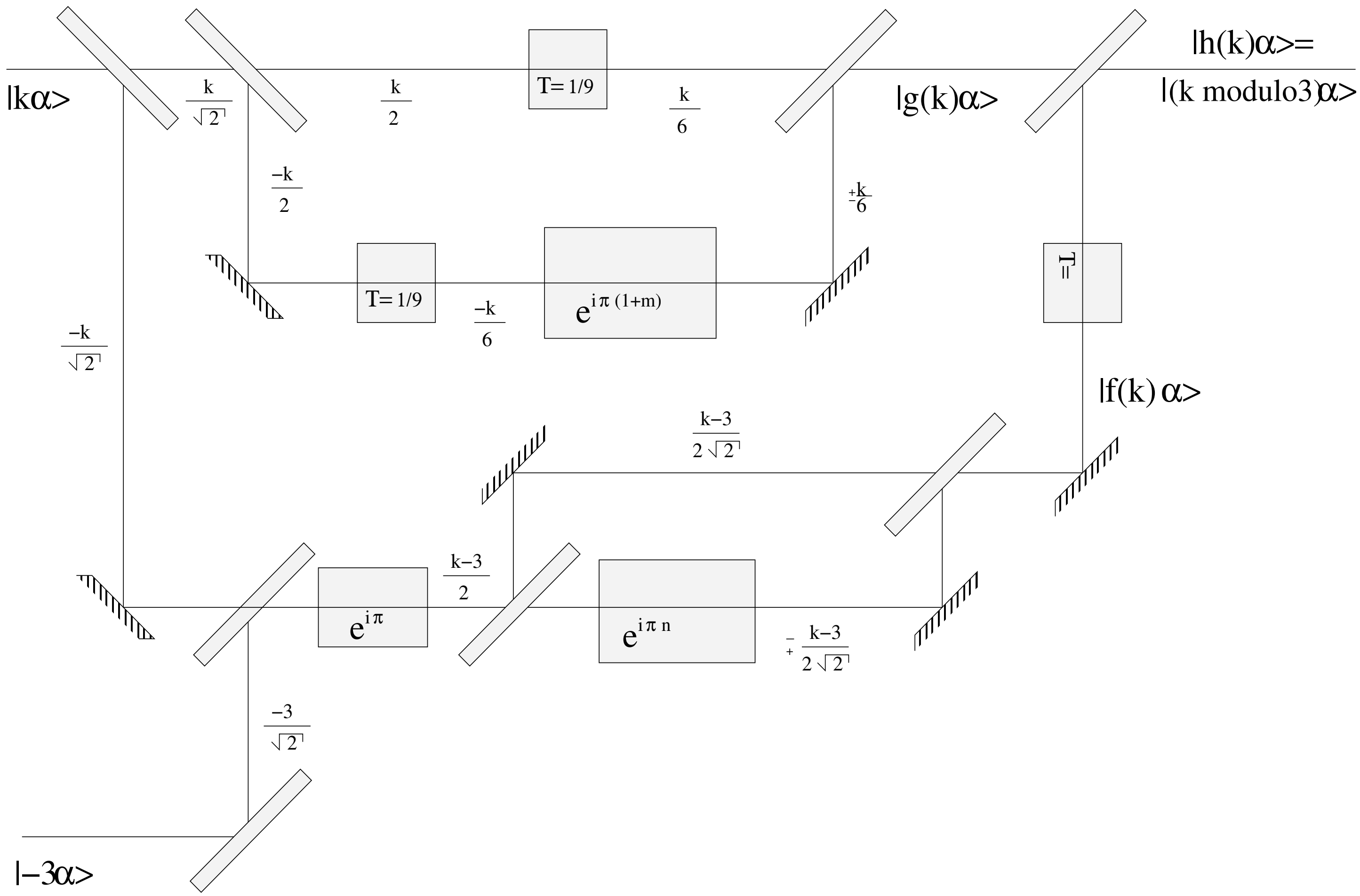


Figure 4.15: Gate that gives out $const \cdot k\ mod3$, k=0,1,2,3,4,5.

4.2.5 Multiplication-gate

Multiplication is one of the fundamental tasks we would like a computer to perform. The goal for a multiplication-gate is to multiply the two or more input signals, i.e. the two input signals, k and j, initialize an output of magnitude $const \cdot k \cdot j$, where the constant may be arbitrary. We may adjust the output at a later point. The configuration described below will produce an output $\frac{kj}{4}$. We experience loss of intensity, but this may be fixed later with an amplifier (see Section 4.2.1).

The following gate largely relies on 50/50 beam-splitters as described in Section 1.2.4. Non-linear crystals are also used to achieve the necessary phase shifts of the coherent states. Some of the crystals simply affect the coherent state by a phase shift of π in the complex plane, while others are "pumped" by the input signal, $|j\alpha>$, and give a phase shift depending on the intensity $|j\alpha|^2$. The phase shift is π when $j = 0$ and 2π when $j \neq 0$.

This is expressed mathematically by

$$m = \begin{cases} 0 & for\ pump = |j| = 0 \ , \\ 1 & for\ pump = |j| > 0 \ . \end{cases} \tag{4.28}$$

The input $|k\alpha\rangle$ is then transformed, by the crystal, to $|ke^{i\pi(m+1)}\alpha\rangle$.
The multiplication-gate makes use of serial addition, where k is the primary input and j is the control input. If $j = 2$, k is added to itself to obtain $2k$, if $j = 1$, k remains unchanged, and if $j = 0$, k is subtracted from itself to obtain 0. The result of the manipulations now becomes

$$|k\alpha>|j\alpha>|2\alpha> \longrightarrow |(j-2)\alpha> \begin{cases} |0> & for\ j = 0 \ , \\ |\frac{k}{4}\alpha> & for\ j = 1 \ , \\ |\frac{2k}{4}\alpha> & for\ j = 2 \ , \end{cases} \tag{4.29}$$

when we neglect output signals of no importance to us.

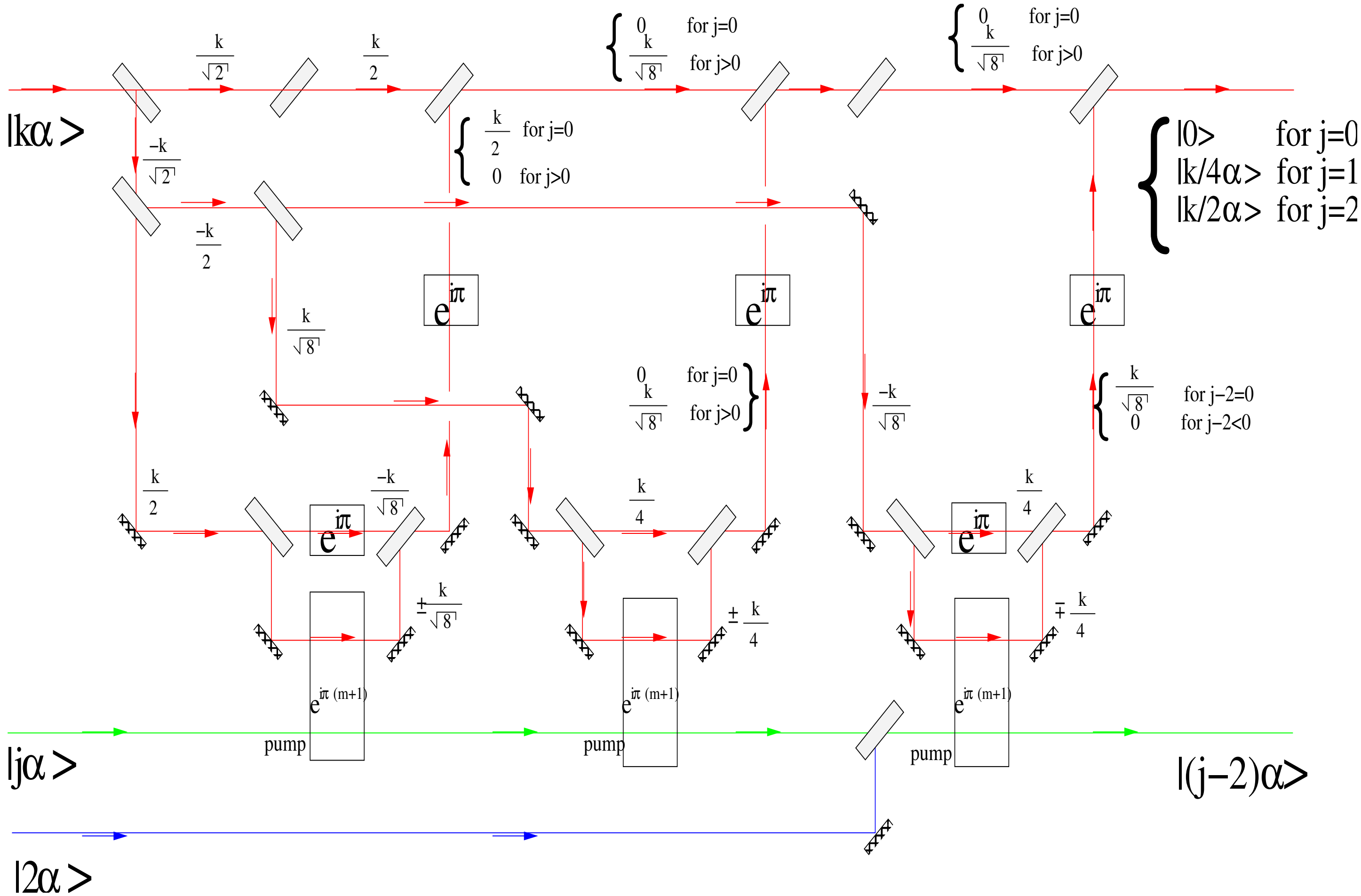


Figure 4.16: Gate that produces kj/4 from two inputs k and j.

4.3 Elementary Logic Gates

4.3.1 Minimum-gate

As stated in Section 2.2.5, there are three possible product-type binary operators, and they are all equally well suited (Ref. [8]). Because of our wish to copy classical binary computing, we will choose the MIN-function to act as a trinary AND-function. The MIN-function is represented by the map

$$\begin{array}{c|ccc} j\backslash k & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 2 \end{array} , \tag{4.30}$$

and we can see that the binary-input AND-gate is included as a special case where $k, j < 2$.
In the design of this trinary-input AND-gate, we have assumed that it is possible to create a medium that has a non-constant transmittivity. We would like the medium's transmittivity, T, to be dependent of the intensity of a pump-laser such that

$$T = \begin{cases} 1 & for \quad I_{pump} > \frac{1}{2} \ , \\ \frac{1}{2} & for \quad I_{pump} = \frac{1}{2} \ , \\ 0 & for \quad I_{pump} < \frac{1}{2} \ . \end{cases} \tag{4.31}$$

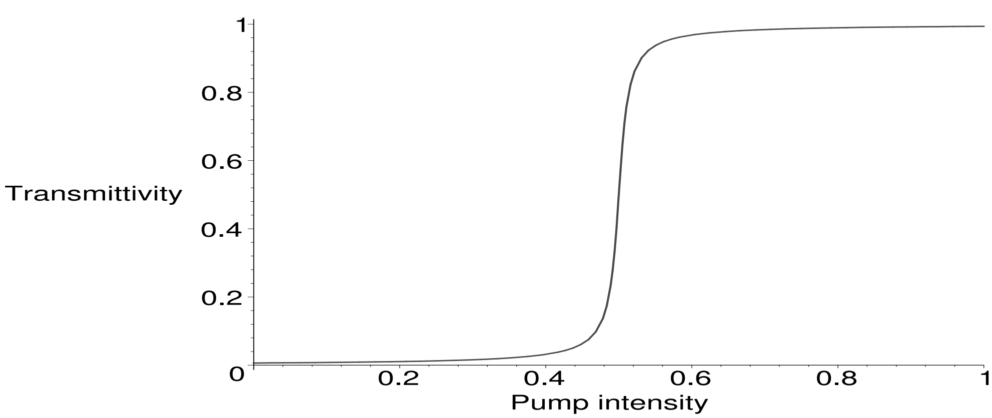


Figure 4.17: Transmittivity as a function of the pump intensity $T(I_{pump}) = \frac{\arctan(100\ I_{pump} - 50) + \frac{\pi}{2}}{\pi}$.

Note that the intensity of a coherent state is defined as $I = \langle\alpha|\hat{a}\hat{a}^{\dagger}|\alpha\rangle = |\alpha|^2$. With such a medium we can realize the minimum-gate as shown in Figure 4.18.

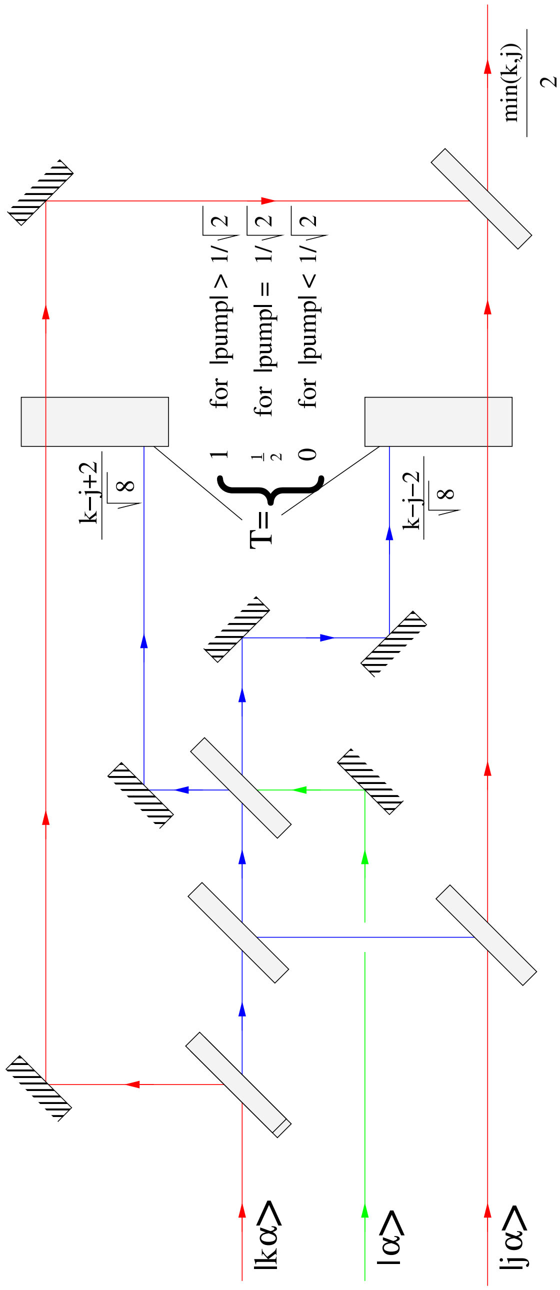


Figure 4.18: Realization of the minimum-gate.

4.3.2 Maximum-gate

The classical binary-input OR-gate may be viewed as a gate which puts out the larger of the two inputs. If following this procedure, a trinary-input OR-gate would produce the following map.

$$\begin{array}{c|ccc} j\backslash k & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array} \quad . \tag{4.32}$$

We see that the binary-input map is included as the case where $k, j < 2$. As we have seen, it was possible to create the trinary-input max-gate from a nonlinear medium and some beam-splitters. The AND-gate was viewed as a gate putting out the least of the two inputs. It is now easy to design the max-gate, since we just have to put out the signal MIN don't put out. The scheme will be exactly the same, except for the third input-state, which for the MIN-gate was $|\alpha\rangle$. The third input-state for the max-gate will be $|-\alpha\rangle$, and we will keep the non-linear medium with the property

$$T = \begin{cases} 1 & for \quad I_{pump} > \frac{1}{2} \ , \\ \frac{1}{2} & for \quad I_{pump} = \frac{1}{2} \ , \\ 0 & for \quad I_{pump} < \frac{1}{2} \ , \end{cases} \tag{4.33}$$

where T is the transmittivity as a function of pump intensity.
The max-gate may be realized as shown in Figure 4.19.

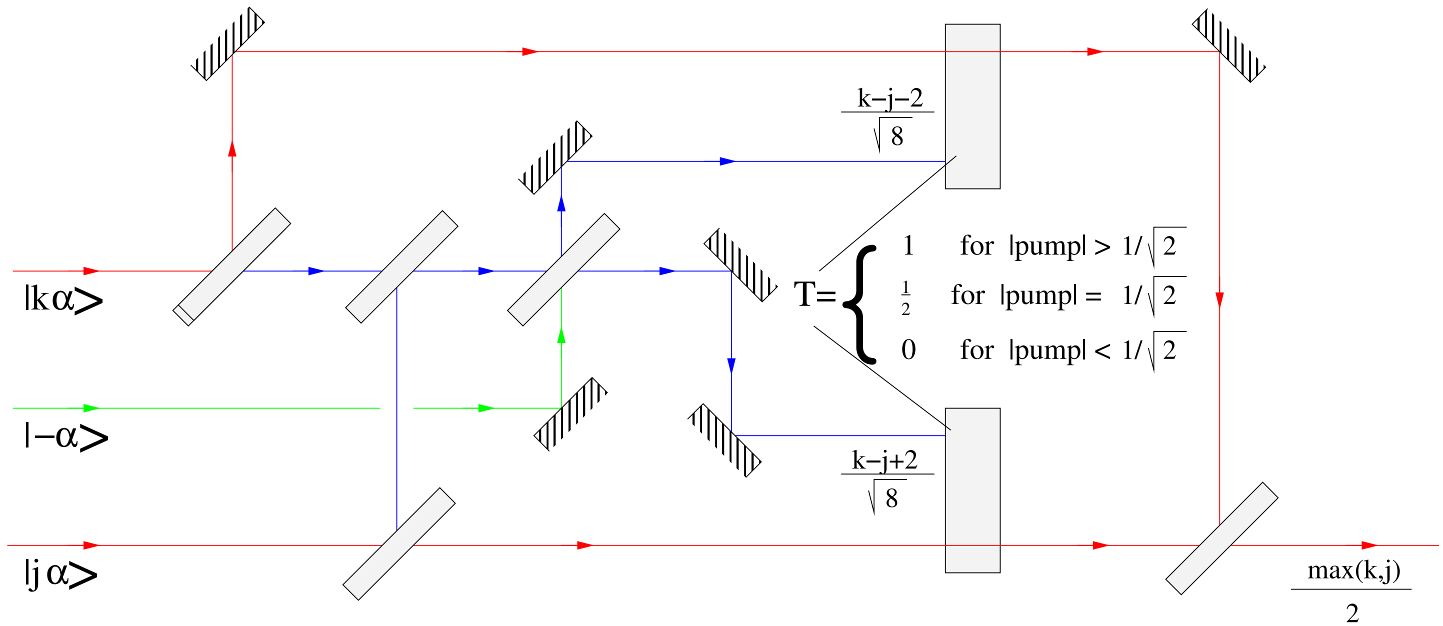


Figure 4.19: Realization of the maximum-gate.

4.3.3 Literal-gate

As earlier stated, we will need a gate that performs the operation *literal*, i.e. for an input signal X and two trinary values $a \leq b$ we get

$$X(a,b) = \begin{cases} 0 & X \notin \{a,b\} \ , \\ 2 & X \in \{a,b\} \ , \end{cases} \tag{4.34}$$

where $\{a,b\}$ denotes an interval of logic values.
First it will be convenient to create a gate, *half-literal*, that gives an output dependent on which is larger, X or a. If we denote the output c, we get

$$c = \begin{cases} 0 & X < a \ , \\ 2 & X \geq a \ . \end{cases} \tag{4.35}$$

This gate will be closely related to the minimum- and maximum-gates, since they also decide which is the largest of the two inputs. We will use the same non-linear media as we used when designing those two gates, but we get into trouble whenever $X = a$ since the transmittivity of the media in such a case is one half. We solve this problem by exchanging the input parameter $|\alpha\rangle$ by another input parameter $|(1+\epsilon)\alpha\rangle$, where $\epsilon \approx 0.5$. In this way we will never get in a situation where $T = 1/2$. The gate in Figure 4.20 will produce the desired output.

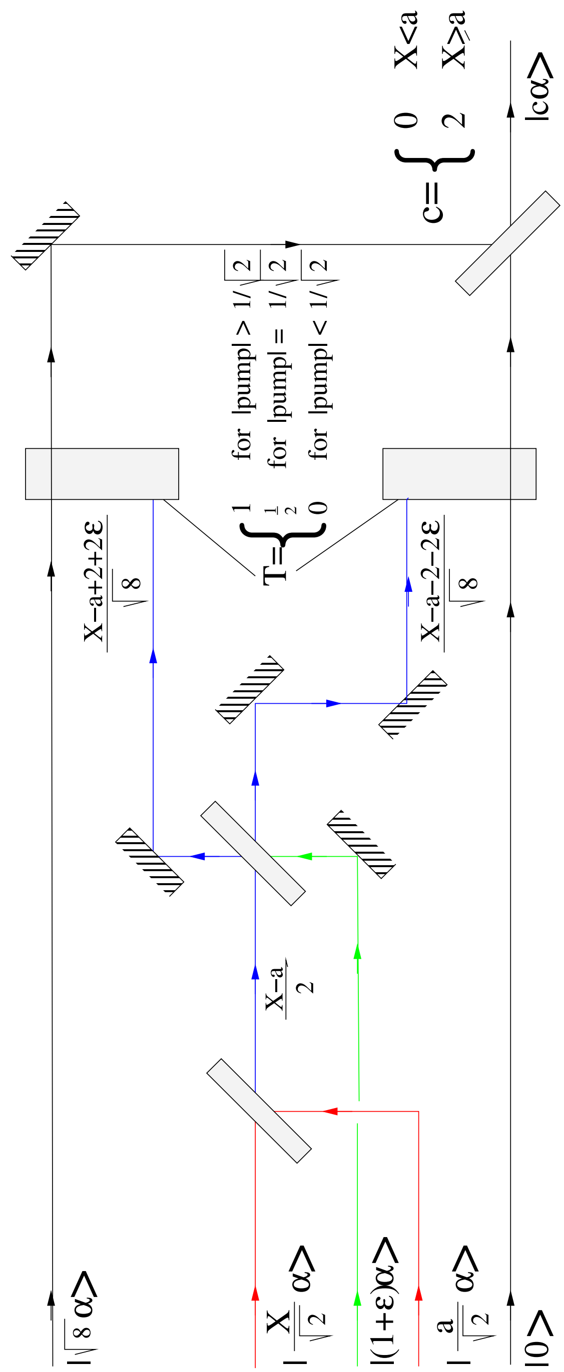


Figure 4.20: Gate that decides which is larger, X or a, and returns one of the values 2 and 0 depending of the answer.

We will now denote the half-literal-gate in Figure 4.20 by the symbol

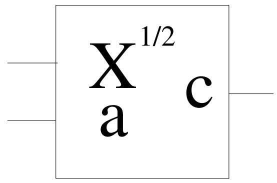


Figure 4.21: Symbol for the gate in Figure 4.20.

We may in the same fashion rather easily construct a gate that performs the literal operation. The half-literal gate is included as an important part of the literal-gate. The output of the half-literal will be either 0 or 2 depending on the size of X compared to a. If X is less than a, the half-literal produces the value 0, and the output of the literal-gate has to be 0. On the other hand, if X is larger than or equal to a, the half-literal produces a 2, and the output of the literal gate may be either 0 or 2 depending on the size of X compared to the size of b (0 if $X > b$ and 2 if $X \leq b$). The literal-gate may be realized as shown in Figure 4.23. We will denote the literal-gate by the symbol

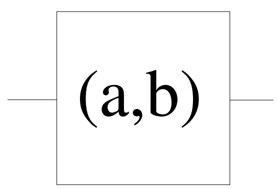


Figure 4.22: Symbol for the gate in Figure 4.23.

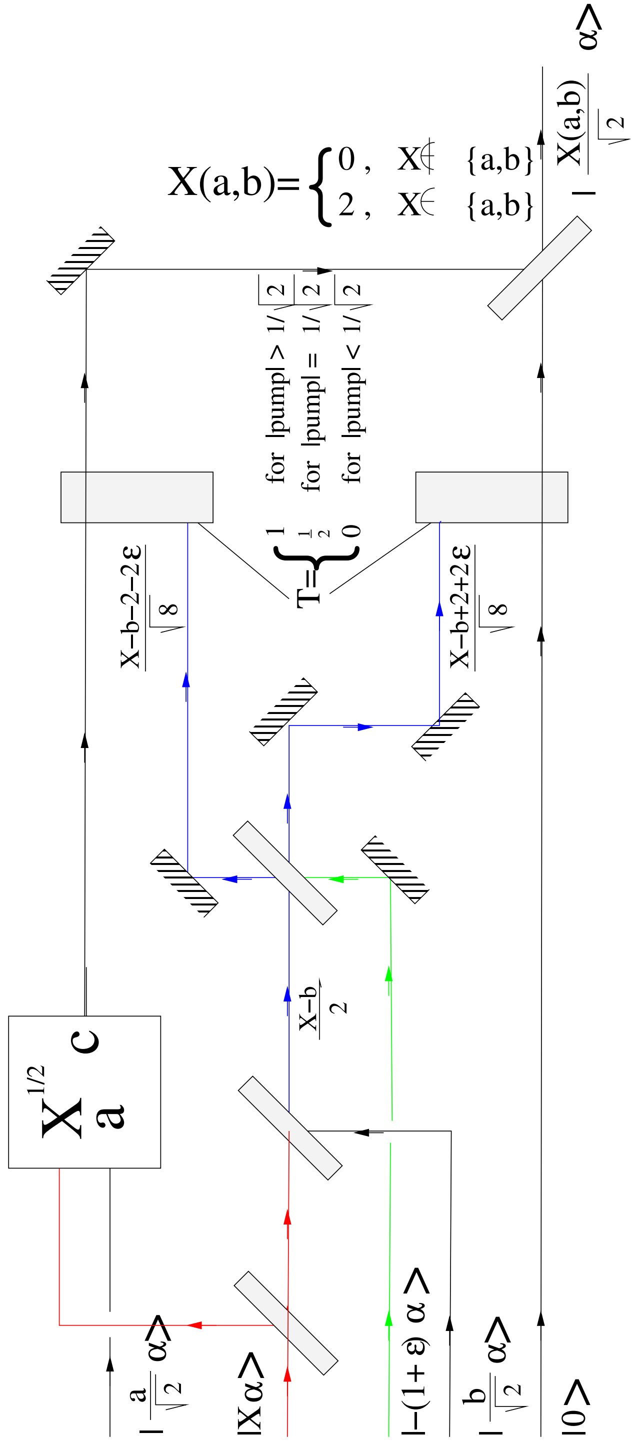


Figure 4.23: Realization of the literal-gate.

4.3.4 Cycle-gate

The cycle-gate is a unary operator and will give the truth-table in Table 4.1.

Table 4.1: Truth-table for the cycle-gate.

X	X^+
0	1
1	2
2	0

As seen in the truth-table, we will denote the cycle operation with a $^+$, meaning $+1$ $mod3$. We could also have defined a negative cycle operation $^-$ meaning -1 $mod3$, but since two cycle operations achieve the same result as one negative cycle operation, this will not be done.

We will assume that there exists a non-linear crystal with the transmittivity property

$$T = \begin{cases} 0 & pump < \alpha \ , \\ \frac{1}{\sqrt{2}} & pump \geq \alpha \ , \end{cases} \tag{4.36}$$

i.e. an input beam of intensity I has the intensity $T^2 \cdot I$ when exiting the crystal.
As seen in Figure 4.24, the pump intensity will be one fourth of the input intensity[5], $|k\alpha|^2$. This means that the crystal is only transparent when the input trit, k, is 2 or higher[6]. When the crystal is transparent, it lets the beam of intensity $9|\alpha|^2$ through, and this beam may now interfere with the modified input.

[5] We recall that the intensity of a coherent state, $|x\alpha\rangle$, is given by $\langle x\alpha|\hat{a}^\dagger\hat{a}|x\alpha\rangle = x^2 \cdot |\alpha|^2$ (x^2 in sloppy notation).

[6] In theory, the input trit can never become larger than 2. It may however come in handy to remember that this gate also may accept intensities that lies beyond the limits of trinary algebra.

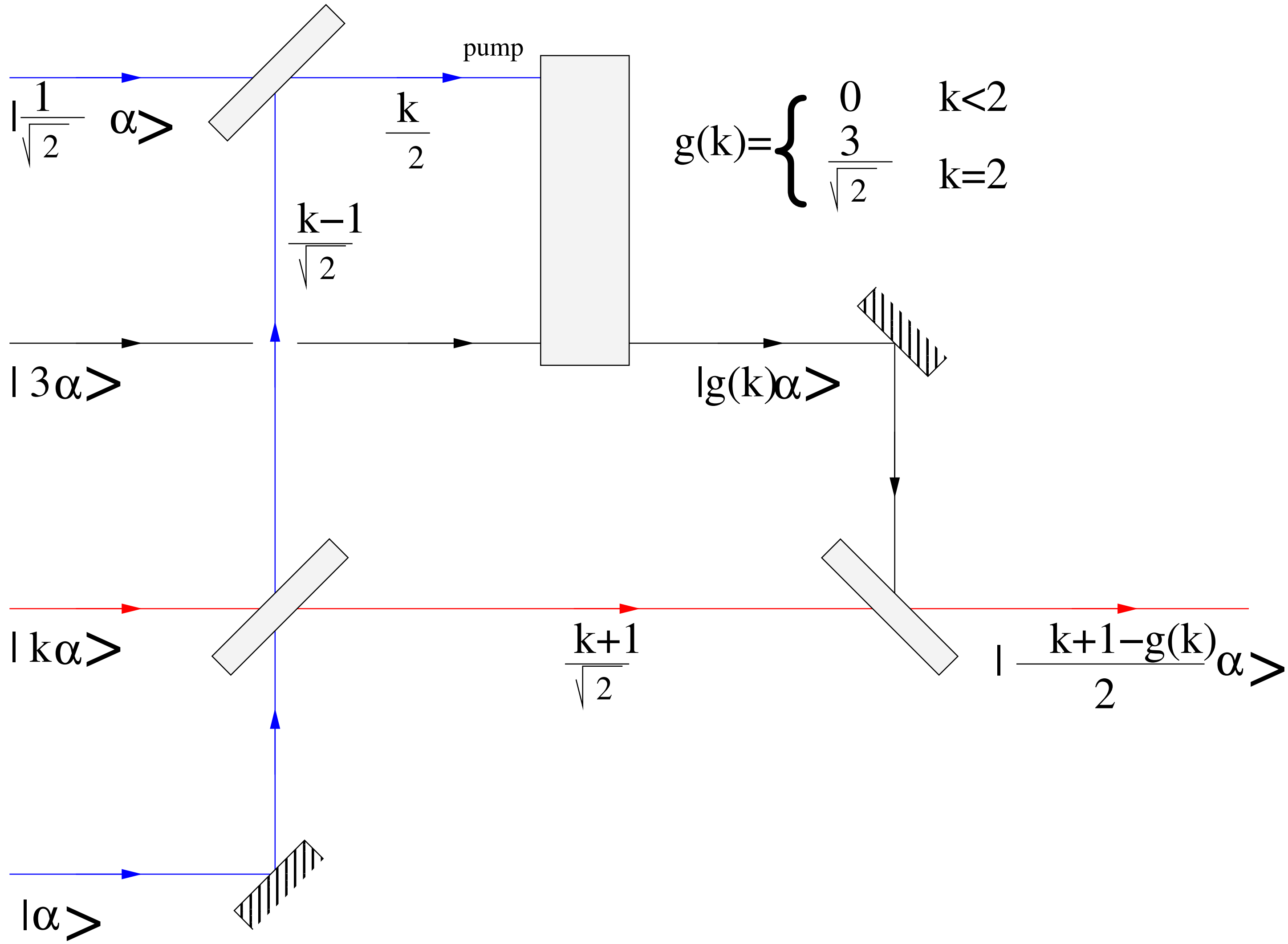


Figure 4.24: Schematic drawing of a cycle-gate.

4.3.5 Inverter-gate

The inverter-gate is a unary operator and will give the truth-table in Table 4.2.

Table 4.2: Truth-table for the inverter-gate.

X	X
0	2
1	1
2	0

As seen in the table, we have adopted the complement notation from Boolean algebra. If we take a look at many-valued logic, 0 may represent false, 1 possibly and 2 truth. The negation of this will become what is shown in Table 4.2. The negation, or inversion, of false is truth, of possible is possible and of truth is false.

The realization of the inverter is quite simple. The input, which is either 0,1 or 2, is subtracted from a constant input of intensity 2. The result, $2 - k$, is exactly the negation as described above.

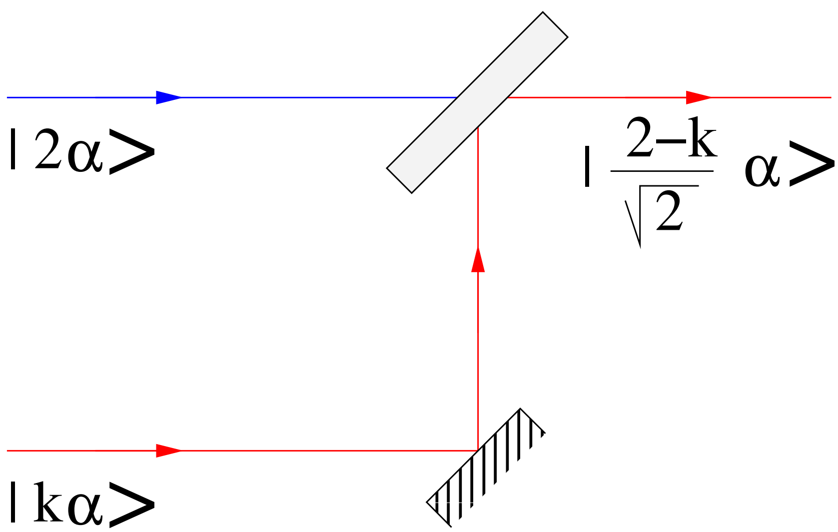


Figure 4.25: Schematic drawing of an inverter-gate.

Final remarks

The reason for writing this thesis, was to propose an efficient scheme for optical quantum computing. As it turned out during the process, the subject is too large to be covered within the time-scale of a cand.scient. degree. We therefore have to be satisfied with the small taste we managed to get, of an interesting and unexplored area of modern physics and computer science.
As was stated in the preface, we may in no way regard this thesis as finished. It should be regarded as a stepping stone for further study.
Two of the most significant problems, that need to be solved are:

1. To be able to exploit the great assets of quantum computing, we need quantum gates that are capable of parallel processing, e.g. an AND-gate that may perform the operation

$$|1\rangle AND(\alpha_1|0\rangle + \beta_1|1\rangle) = \alpha_2|0\rangle + \beta_2|1\rangle, \tag{4.37}$$

where the ket-vectors denote logic states, 0 or 1, and we have suppressed outputs of no interest.

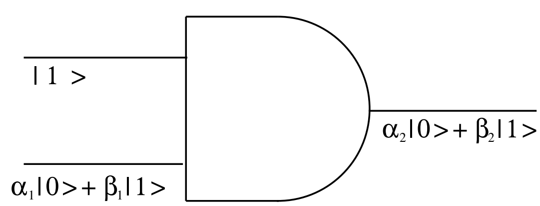


Figure 4.26: Drawing of the process in Eq. (4.37).

At present time, in the descriptions given in Chapter 4, not all gates proposed are able to perform these kinds of parallel processes. As previously mentioned, this is especially the case where non-linear crystals are used. This is a problem that needs to be solved in order to create effective quantum computers.

2. When rejecting outputs of the quantum-gates, as they are of no interest, we experience intensity-loss at a large scale. The loss differs from gate to gate, but it generally represents a severe problem. As we have decided that the logic states are represented by coherent states of corresponding intensity, a logic state is actually altered as a corresponding coherent state looses intensity. Of course this problem is easily solved in simple circuits. However, the problem grows as the complexity of the circuits grow. Since most of the gates are quite fastidious, regarding the intensity of the input signals, we will have to restore the original signal, i.e. amplify the present signal, at frequent occasions. This problem needs to be solved in order to create large, complex quantum circuits.

One may wonder, whether it turned out to be a good idea or not, to develop the quantum gates in a trinary environment. As we have seen, the algebraic expressions grow quite complicated, and the minimization procedures are not fully developed. It may have been simpler to develop a quantum-optical scheme of computing within the already existing, Boolean algebra. Note, however, that most of the quantum-gates proposed in this thesis include the binary-logic gates, e.g. if going back to binary logic, MAX acts as OR and MIN acts as AND. One may even choose which state (of the three trinary states, $|0\rangle$, $|1 \cdot \alpha\rangle$ and $|2 \cdot \alpha\rangle$) we shall remove. The easiest will perhaps be to remove the possible-state, i.e. 1 ($|1 \cdot \alpha\rangle$). Then $|0\rangle$ will denote logic 0 and $|2 \cdot \alpha\rangle$ will denote logic 1 (in binary logic). The inverter-gate will then be equivalent to logic NOT, and all the basic logic gates of Boolean algebra are recalled.

Despite the problems discovered during the writing of this thesis, the author strongly believes that the quantum-optical approach to quantum computing is the most practical approach. The value of higher order logic may as stated be discussed, but even with the present Boolean algebra or the existing quantum-logic, the author believes that the advantages of quantum-optical states as logic states, in the long run will ensure that quantum-optical computers emerge victorious from the struggle against present computing regimes.
There might however still be undiscovered physical systems or properties of already known systems, which would be even more fit to solve the computational tasks.

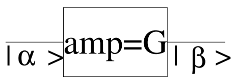
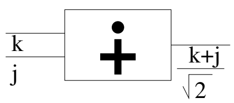
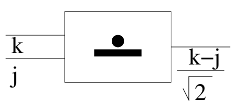
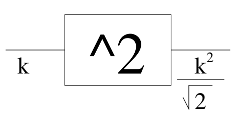
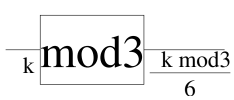
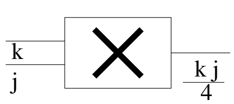
Appendix A

Quantum-optical gates; an overview

In Chapter 4, there is proposed a variety of quantum-optical gates. In the tables below, an overview of these is provided, together with each gate's specific properties, i.e. symbol, required input and output.

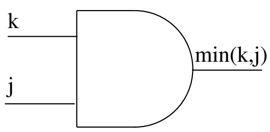
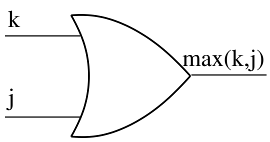
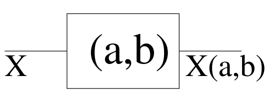
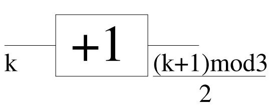
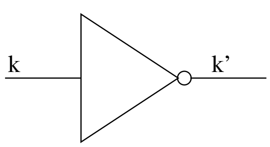
A.1 Specialized Gates

Table A.1: Overview of symbol, required input and output of specialized quantum-optical gates proposed in Chapter 4.

<table>
<tr><th>Symbol</th><th>Input</th><th>Output</th><th>Comment</th></tr>
<tr><td>Amplifier (Section 4.2.1).
</td><td>$|\alpha\rangle$, where $\langle\alpha|\hat{a}^\dagger\hat{a}|\alpha\rangle = n$</td><td>$|\beta\rangle$, where $\langle\beta|\hat{a}^\dagger\hat{a}|\beta\rangle = m + G \cdot n$</td><td>The spontaneous emission field contributes with m photons. The amplifier field contributes with $G \cdot n$ photons.</td></tr>
<tr><td>Adder (Section 4.2.2).
</td><td>$|k \cdot \alpha\rangle$ and $|j \cdot \alpha\rangle$, where $k, j \in \mathbb{C}$.</td><td>$\frac{1}{\sqrt{2}}|(k+j) \cdot \alpha\rangle$.</td><td>Since k may be any complex number, the adder is actually an analog computer.</td></tr>
<tr><td>Subtractor (Section 4.2.2).
</td><td>$|k \cdot \alpha\rangle$ and $|j \cdot \alpha\rangle$, where $k, j \in \mathbb{C}$.</td><td>$\frac{1}{\sqrt{2}}|(k-j) \cdot \alpha\rangle$.</td><td>Since k may be any complex number, the subtractor is actually an analog computer. The adder together with the modulo3-gate is also called MODSUM and may be regarded as an equivalent to binary-logic OR.</td></tr>
<tr><td>Square-gate (Section 4.2.3).
</td><td>$|k \cdot \alpha\rangle$, where $k \in \{0, 1, 2\}$.</td><td>$|\frac{k^2}{\sqrt{2}}\alpha\rangle$.</td><td>The gate may accept other inputs, but the outputs must be calculated by analyzing the schematic drawing of the gate.</td></tr>
<tr><td>Modulo3-gate (Section 4.2.4).
</td><td>$|k \cdot \alpha\rangle$, where $k \in \{0, 1, 2\}$.</td><td>$|\frac{k \text{ modulo } 3}{6}\alpha\rangle$.</td><td>Some gates may put out signals that are outside the boundaries of the trinary algebra. We may then have to modify the output signal by the modulo3-gate.</td></tr>
<tr><td>Multiplication-gate (Section 4.2.5).
</td><td>$|k \cdot \alpha\rangle$ and $|j \cdot \alpha\rangle$, where $k, j \in \{0, 1, 2\}$.</td><td>$|\frac{k \cdot j}{4}\alpha\rangle$.</td><td>The multiplication-gate together with the modulo3-gate may be regarded as an equivalent to the binary-logic XOR.</td></tr>
</table>

A.2 Elementary Logic Gates

Table A.2: Overview of symbol, required input and output of elementary quantum-optical logic gates proposed in Chapter 4.

Symbol	Input	Output	Comment
Minimum-gate (Section 4.3.1) 	$\|k\alpha\rangle$ and $\|j\alpha\rangle$, where $k, j \in \{0, 1, 2\}$.	$\|\frac{min(k,j)}{2}\alpha\rangle$	The minimum-gate may be interpreted as an AND-gate. For the trinary values 0 and 1, the gate behaves exactly like binary logic AND.
Maximum-gate (Section 4.3.2) 	$\|k\alpha\rangle$ and $\|j\alpha\rangle$, where $k, j \in \{0, 1, 2\}$.	$\|\frac{max(k,j)}{2}\alpha\rangle$	The maximum-gate may be interpreted as an OR-gate. For the trinary values 0 and 1, the gate behaves exactly like binary logic OR.
Literal (Section 4.3.3) 	$\|a\alpha\rangle$, $\|b\alpha\rangle$ and $\|X\alpha\rangle$, where $a, b, X \in \{0, 1, 2\}$.	$\|\frac{X(a,b)}{\sqrt{2}}\alpha\rangle$, where $X(a,b) = \begin{cases} 0 & X \notin [a,b] \\ 2 & X \in [a,b] \end{cases}$	
Cycle-gate (Section 4.3.4) 	$\|k\alpha\rangle$, where $k \in \{0, 1, 2\}$.	$\|\frac{(k+1)mod3}{2}\alpha\rangle$.	
Inverter-gate (Section 4.3.5) 	$\|k\alpha\rangle$, where $k \in \{0, 1, 2\}$.	$\|\frac{2-k}{\sqrt{2}}\alpha\rangle$.	The inverter-gate may be regarded as a trinary NOT-gate.

Appendix B

Source Codes

B.1 Verification of the Huntington Postulates

In this section some of the Huntington postulates are verified for a trinary algebra, by testing all possibilities.

B.1.1 Program to check postulate 4a

```
#include<iostream.h>



int max(int a, int b){
  int c;
  if (a>=b) c=a;
  else c=b;
  return c;
}

int min(int a, int b){
  int c;
  if (a<=b) c=a;
  else c=b;
  return c;
}

void main(){

  int i,j,k;
  int result1, result2;


  cout << "i,j,k" << "    " << "x(y+z)" << "   " << "(xy)+(xz)" << "\n";
  for(i=0;i<3;i++){
    for(j=0;j<3;j++){
      for(k=0;k<3;k++){
result1=min(i,max(j,k));
result2=max(min(i,j),min(i,k));
cout << i << "," << j << "," << k<<  "    " << result1 << "   " << result2 << "\n";
      }
    }
```

```
  }
}
```

B.1.2 Program to check postulate 4b

```
#include<iostream.h>



int max(int a, int b){
  int c;
  if (a>=b) c=a;
  else c=b;
  return c;
}

int min(int a, int b){
  int c;
  if (a<=b) c=a;
  else c=b;
  return c;
}

void main(){

  int i,j,k;
  int result1, result2;


  cout << "i,j,k" << "    " << "x+(yz)" << "   " << "(x+y)(x+z)" << "\n";
  for(i=0;i<3;i++){
    for(j=0;j<3;j++){
      for(k=0;k<3;k++){
result1=max(i,min(j,k));
result2=min(max(i,j),max(i,k));
cout << i << "," << j << "," << k << "    " << result1 << "   " << result2 << "\n";
      }
    }
  }
}
```

B.2 Proofs of the Theorems of Boolean Algebra

In this section some of the theorems from Boolean algebra are shown to apply to trinary algebra as well.

B.2.1 Program to prove theorem 5a

```
#include<iostream.h>



int max(int a, int b){
  int c;
  if (a>=b) c=a;
```

```
  else c=b;
  return c;
}

int min(int a, int b){
  int c;
  if (a<=b) c=a;
  else c=b;
  return c;
}

int comp(int a){
  int c=2-a;
  return c;
}


void main(){

  int i,j,k;
  int result1, result2;


  cout << "i,j" << "    " << "(x+y)'" << "   " << "x'y'" << "\n";
  for(i=0;i<3;i++){
    for(j=0;j<3;j++){
      result1=comp(max(i,j));
      result2=min(comp(i),comp(j));
      cout << i << "," << j  << "    " << result1 << "   " << result2 << "\n";
    }
  }
}
```

B.2.2 Program to prove theorem 5b

```
#include<iostream.h>



int max(int a, int b){
  int c;
  if (a>=b) c=a;
  else c=b;
  return c;
}

int min(int a, int b){
  int c;
  if (a<=b) c=a;
  else c=b;
  return c;
}

int comp(int a){
  int c=2-a;
  return c;
}


void main(){
```

```
  int i,j,k;
  int result1, result2;


  cout << "i,j" << "    " << "(x+y)'" << "   " << "x'y'" << "\n";
  for(i=0;i<3;i++){
    for(j=0;j<3;j++){
      result1=comp(min(i,j));
      result2=max(comp(i),comp(j));
      cout << i << "," << j  << "    " << result1 << "   " << result2 << "\n";
    }
  }
}
```

B.3 Trinary Sum-type Functions

There are 3^9 possible two-variable three-valued functions. Sum-type functions, denoted by $+$, are associative, commutative and they give $x + 0 = x$. The following C++ program calculates all possible functions and prints the sum-type functions to the screen.

```
//Calculates all possible two-input three-valued functions and prints
//to the screen those which are sum-type.

#include <iostream.h>


void main(){
  int vector1[9];
  int array[3][3];
  int flag1=0;
  int flag2=0;
  int flag3=0;
  int commutative1, commutative2;
  int associative1, associative2;
  int i,j,k,l;
  int number=0;

  //Initializing array
  for(i=0;i<3;i++)
    for(j=0;j<3;j++)
      array[i][j]=0;

  //Initializing vector1
  for(i=0;i<9;i++)
    vector1[i]=0;



  //generating 3^9 different functions
  for(i=0;i<100000;i++){
    flag1=0;
    flag2=0;
    flag3=0;

    for(j=0;j<8;j++){
      if(vector1[j]==3){
```

```
vector1[j]=0;
vector1[j+1]++;
      }
    }


    //converting vector->array
    for(j=0;j<3;j++){
      array[0][j]=vector1[j];
      array[1][j]=vector1[j+3];
      array[2][j]=vector1[j+6];
    }


    //Checking 0+x=x
    for(j=0;j<3;j++)
      if(array[0][j]==j)
flag1++;
    if(flag1==3)
      flag1=1;
    else flag1=0;


    //Checking commutativity if x+0=x
    if(flag1==1)
      for(j=0;j<3;j++)
for(k=0;k<3;k++)
  if(array[j][k]==array[k][j])
    flag2++;
    if(flag2==9)
      flag2=1;
    else flag2=0;

    //Checking associativity if commutative
    if(flag2==1)
      for(j=0;j<3;j++)
for(k=0;k<3;k++)
  for(l=0;l<3;l++)
    if(array[j][array[k][l]]==array[array[j][k]][l])
      flag3++;
    if(flag3==27)
      flag3=1;
    else flag3=0;

    //Printing the function if commutative and associative
    if(flag3==1){
      number++;
      cout << "     |0  1  2\n" << "   __________\n" << "   0 |"
   << array[0][0] << "  " << array[0][1] << "  " << array[0][2]
   << "\n"  << "   1 |"<< array[1][0] << "  " << array[1][1]
   << "  " << array[1][2] << "\n" << "   2 |" << array[2][0]
   << "  " << array[2][1] << "  " << array[2][2] << "\n\n"
   << "       ("  << number << ")\n\n\n";
    }


    if(vector1[8]==3) break;
    vector1[0]++;
  }
}
```

Appendix C

Linear Amplifier

C.1 Chaotic Light

When blackbody radiation is passed through some filter, the radiation may be altered in such a way that it is no longer true blackbody radiation. The light may be directional rather than isotropic, partially polarized rather than unpolarized, spatially inhomogeneous, and it may have a spectral distribution different from that given by the Planck distribution (Ref. [1]). Radiation derivable from blackbody radiation by any linear filtering process is called *chaotic radiation*.

The density operator of a blackbody radiation field may be represented by coherent states as

$$\hat{\rho} = \int \phi(\{\nu\})|\{\nu\}\rangle\langle\{\nu\}|d\{\nu\} \ , \tag{C.1}$$

with the phase-space functional given by

$$\phi(\{\nu\}) = \int \langle -\{u\}|\hat{\rho}|\{u\}\rangle \prod_{\mathbf{k},s} \left[e^{|\nu_{\mathbf{k}s}|^2+|u_{\mathbf{k}s}|^2} e^{u^*_{\mathbf{k}s}\nu_{\mathbf{k}s}-u_{\mathbf{k}s}\nu^*_{\mathbf{k}s}} \frac{d^2u_{\mathbf{k}s}}{\pi^2} \right] \ . \tag{C.2}$$

It may now be shown that

$$\phi(\{\nu\}) = \prod_{\mathbf{k},s} \frac{1}{\pi\langle n_{\mathbf{k}s}\rangle} e^{-|\nu_{\mathbf{k}s}|^2/\langle n_{\mathbf{k}s}\rangle} \ , \tag{C.3}$$

where $\langle n_{\mathbf{k}s}\rangle$ is the mean photon number of mode $\mathbf{k}, s$,

$$\langle n_{\mathbf{k}s}\rangle = \frac{1}{e^{\hbar\omega/k_BT}-1} \ . \tag{C.4}$$

If the effect of the filter is represented by a homogeneous linear transformation of the field, then the Gaussian statistics of the Fourier amplitudes of blackbody radiation will be retained by chaotic radiation (Ref. [1]), and the phase space functional $\phi(\{\nu\})$ describing the state will again be Gaussian. However, $\phi(\{\nu\})$ is now of the general multivariate Gaussian form

$$\phi(\{\nu\}) = \frac{1}{det(\pi\mu)} e^{-\mathbf{v}^\dagger \mu^{-1} \mathbf{v}} \ , \tag{C.5}$$

rather than of the particular form of Eq. (C.3). $\mathbf{v}$ stands for the column matrix formed by the set of complex amplitudes $\{\nu\}$, and μ is the covariant matrix with elements $\mu_{\mathbf{k}s,\mathbf{k}'s'} = \langle \nu^*_{\mathbf{k}s} \nu_{\mathbf{k}'s'} \rangle_\phi$. The exponent is a general bilinear functional of the set $\{\nu\}$, and it can be written as

$$\mathbf{v}^\dagger \mu^{-1} \mathbf{v} = \sum_{\mathbf{k},s} \sum_{\mathbf{k}',s'} \nu^*_{\mathbf{k}s} \mu^{-1}_{\mathbf{k}s,\mathbf{k}'s'} \nu_{\mathbf{k}'s'} \ . \tag{C.6}$$

Since μ^{-1} is not necessarily diagonal, the phase space functional $\phi(\{\nu\})$ no longer factorizes into a product of separate mode distributions, and the different mode amplitudes, $\nu_{\mathbf{k}s}$, are not statistically independent in general. However, if we integrate over all modes except one, in order to derive the phase space distribution of the $\mathbf{k}', s'$ mode, we find that

$$\phi(\{\nu_{\mathbf{k}'s'}\}) = \frac{1}{\pi \langle n_{\mathbf{k}'s'} \rangle} e^{-|\nu_{\mathbf{k}'s'}|^2 / \langle n_{\mathbf{k}'s'} \rangle} \ . \tag{C.7}$$

where the average photon number, $\langle n_{\mathbf{k}'s'} \rangle$ no longer is given by Eq. (C.4).

Let us now calculate the average value of the thermal field. We will evaluate the integral

$$\int\limits_{plane} \nu \phi_s(\nu, \nu^*) d^2\nu \ \sim \ \int\limits_{plane} \nu e^{-|\nu|^2} d^2\nu = 0 \ . \tag{C.8}$$

So the average value of the thermal field is 0. When taking a look at the plot of the integrand in Figure C.1, we can clearly see that it is anti-symmetric. Hence, the integral must be zero, i.e. the average value of the thermal field is zero.

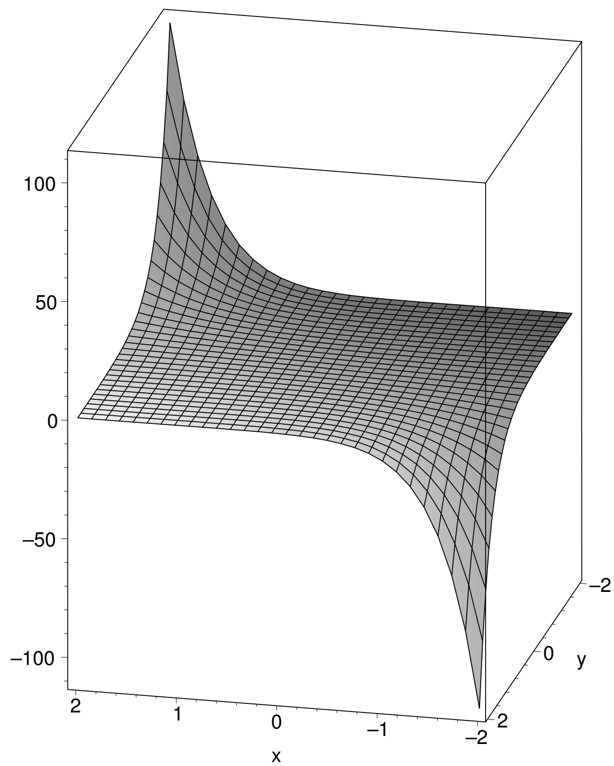


Figure C.1: Integrand of average chaotic field, $\langle \nu \rangle = \int \nu \phi(\nu) d^2\nu$.

C.2 Calculations from Section 4.2.1

C.2.1 Master equation for the amplifier field

As stated in Section 4.2.1, the master equation for the amplifier field is

$$\frac{\partial \hat{\rho}}{\partial t} = -\frac{1}{2}A(\hat{a}\hat{a}^\dagger\hat{\rho} - \hat{a}^\dagger\hat{\rho}\hat{a} + h.c.) - \frac{1}{2}C(\hat{a}^\dagger\hat{a}\hat{\rho} - \hat{a}\hat{\rho}\hat{a}^\dagger + h.c.) \ . \tag{C.9}$$

We may now make use of the relations (Ref. [1])

$$\hat{a}^\dagger|\nu\rangle\langle\nu| = \left(\nu^* + \frac{\partial}{\partial\nu}\right)|\nu\rangle\langle\nu| \ , \tag{C.10}$$

$$|\nu\rangle\langle\nu|\hat{a} = \left(\nu + \frac{\partial}{\partial\nu^*}\right)|\nu\rangle\langle\nu| \ . \tag{C.11}$$

By using Eq. (C.10) and Eq. (C.11) we get

$$\begin{aligned}\hat{a}\hat{a}^\dagger|\nu\rangle\langle\nu| &= (\hat{a}^\dagger\hat{a}+1)|\nu\rangle\langle\nu| \\ &= \left[\nu(\nu^* + \frac{\partial}{\partial\nu})\right]|\nu\rangle\langle\nu| \ ,\end{aligned} \tag{C.12}$$

$$\begin{aligned}-\hat{a}^\dagger|\nu\rangle\langle\nu|\hat{a} &= -\left(\nu^* + \frac{\partial}{\partial\nu}\right)\left(\nu + \frac{\partial}{\partial\nu^*}\right)|\nu\rangle\langle\nu| \\ &= -\left(|\nu|^2 + \nu^*\frac{\partial}{\partial\nu} + \nu\frac{\partial}{\partial\nu^*} + \frac{\partial^2}{\partial\nu\partial\nu^*}\right)|\nu\rangle\langle\nu| \ ,\end{aligned} \tag{C.13}$$

$$\hat{a}^\dagger\hat{a}|\nu\rangle\langle\nu| = \nu\left(\nu^* + \frac{\partial}{\partial\nu}\right)|\nu\rangle\langle\nu| \tag{C.14}$$

and

$$-\hat{a}|\nu\rangle\langle\nu|\hat{a}^\dagger = -|\nu|^2|\nu\rangle\langle\nu| \ . \tag{C.15}$$

We now insert the representation

$$\hat{\rho}(t) = \int \phi(\nu,t)|\nu\rangle\langle\nu|d^2\nu \ , \tag{C.16}$$

of the density operator into the master equation Eq. (C.9), and we replace $\hat{a}$ and $\hat{a}^\dagger$ by the corresponding differential operators as shown in Eq. (C.10) and Eq. (C.11). We then get, after some rearrangements, the equation of motion

$$\begin{aligned}\int \frac{\partial\phi(\nu,t)}{\partial t}|\nu\rangle\langle\nu|d^2\nu = \int \phi(\nu,t)\Bigg\{&-\frac{1}{2}A\left[-\nu^*\frac{\partial}{\partial\nu^*} - \nu\frac{\partial}{\partial\nu} - 2\frac{\partial^2}{\partial\nu\partial\nu^*}\right] \\ &- \frac{1}{2}C\left[\nu^*\frac{\partial}{\partial\nu^*} + \nu\frac{\partial}{\partial\nu}\right]\Bigg\}|\nu\rangle\langle\nu|d^2\nu \ .\end{aligned} \tag{C.17}$$

By formally integrating by parts, with the assumption that $\phi(\nu,t)$ vanishes at infinity faster than any power of ν and ν^*, we can convert the integrand on the right into a product of $|\nu\rangle\langle\nu|$ and a c-number function of ν and ν^*, and we obtain the formula

$$\begin{aligned}&\int |\nu\rangle\langle\nu|\frac{\partial\phi(\nu,t)}{\partial t}d^2\nu = \\ &\qquad\int |\nu\rangle\langle\nu|\left\{-\frac{1}{2}(A-C)\left(\frac{\partial}{\partial\nu}\nu + \frac{\partial}{\partial\nu^*}\nu^*\right) + A\frac{\partial^2}{\partial\nu\partial\nu^*}\right\}d^2\nu \ ,\end{aligned} \tag{C.18}$$

which leads us to the equation

$$\frac{1}{\lambda}\frac{\partial\phi(\nu,t)}{\partial t} = \\ -(N_2-N_1)\left[\frac{\partial}{\partial\nu}(\nu\phi(\nu,t))+\frac{\partial}{\partial\nu^*}(\nu^*\phi(\nu,t))\right]+2N_2\frac{\partial^2\phi(\nu,t)}{\partial\nu\partial\nu^*}\ . \quad \text{(C.19)}$$

C.2.2 Solution of the master equation

Assuming we have the coherent state, $|\nu'\rangle$, whose density matrix, ρ, is given by

$$\rho=\int\phi_s(\nu',t)|\nu'\rangle\langle\nu'|d^2\nu\ . \quad \text{(C.20)}$$

The Fokker-Planck equation for the amplifier field is, for the initial state, $|\nu'\rangle$, (see Eq. (C.19))

$$\frac{1}{\lambda}\frac{\partial\phi_s(\nu,t)}{\partial t} = \\ -(N_2-N_1)\left[\frac{\partial}{\partial\nu}(\nu\phi_s(\nu,t))+\frac{\partial}{\partial\nu^*}(\nu^*\phi_s(\nu,t))\right]+2N_2\frac{\partial^2\phi_s(\nu,t)}{\partial\nu\partial\nu^*}\ . \quad \text{(C.21)}$$

If substituting for

$$\mathscr{C} \equiv -2\lambda(N_2-N_1)\ , \quad \text{(C.22)}$$

$$\bar{n} \equiv \frac{-N_2}{N_2-N_1}\ , \quad \text{(C.23)}$$

we get the equation

$$\frac{\partial\phi_s(\nu,t)}{\partial t}=\frac{\mathscr{C}}{2}\left[\frac{\partial}{\partial\nu}(\nu\phi(\nu,t))+\frac{\partial}{\partial\nu^*}(\nu^*\phi(\nu,t))\right]+\mathscr{C}\bar{n}\frac{\partial^2\phi(\nu,t)}{\partial\nu\partial\nu^*}\ . \quad \text{(C.24)}$$

Since the initial field is coherent,

$$\phi_s(\nu,0)=\delta^2(\nu-\nu')\ , \quad \text{(C.25)}$$

which in the Gaussian representation becomes

$$\phi_s(\nu,t)=\lim_{\epsilon\to 0}\ \frac{1}{\pi\epsilon}\exp\left(\frac{-|\nu-\nu'|^2}{\epsilon}\right)\ . \quad \text{(C.26)}$$

We may now search for a solution of the form

$$\exp\left[-a(t)+b(t)\nu+c(t)\nu^*-d(t)\nu\nu^*\right]\ , \quad \text{(C.27)}$$

subject to the initial conditions

$$
\begin{aligned}
a(0) &= \frac{|\nu'|^2}{\epsilon} + \ln(\pi\epsilon) \ , && \text{(C.28)}\\
b(0) &= \frac{\nu'^*}{\epsilon} \ , && \text{(C.29)}\\
c(0) &= \frac{\nu'}{\epsilon} \ , && \text{(C.30)}\\
d(0) &= \frac{1}{\epsilon} \ . && \text{(C.31)}
\end{aligned}
$$

On substituting Eq. (C.27) into Eq. (C.24) and carrying out the necessary t and ν differentiations, we get

$$
\begin{aligned}
&-\dot{a} + \dot{b}\nu' + \dot{c}\nu'^* - \dot{d}|\nu'|^2 = \\
&\mathscr{C}\left[1 + \bar{n}(bc-d) + (\frac{b}{2} - \bar{n}bd)\nu' + (\frac{c}{2} - \bar{n}cd)\nu'^* - (d - \bar{n}d^2)|\nu'|^2\right] \ . \qquad \text{(C.32)}
\end{aligned}
$$

By comparing terms proportional to ν',ν'^*,$|\nu'|^2$ and unity, we get the following set of differential equations:

$$
\begin{aligned}
\dot{a} &= -\mathscr{C}\left[1 + \bar{n}(bc-d)\right] \ , && \text{(C.33)}\\
\dot{b} &= \mathscr{C}\left(\frac{b}{2} - \bar{n}bd\right) \ , && \text{(C.34)}\\
\dot{c} &= \mathscr{C}\left(\frac{c}{2} - \bar{n}cd\right) \ , && \text{(C.35)}\\
\dot{d} &= \mathscr{C}\left(d - \bar{n}d^2\right) \ . && \text{(C.36)}
\end{aligned}
$$

The solution of these equations subject to the initial conditions (Eq. (C.28)-(C.31)) is given by

$$
\begin{aligned}
a(t) &= \frac{|\nu'|^2 e^{-\mathscr{C}t}}{\bar{n}(1-e^{-t}) + \epsilon e^{-\mathscr{C}t}} + \ln\left\{\pi\left[\bar{n}\left(1 - e^{-\mathscr{C}t}\right) + \epsilon e^{-\mathscr{C}t}\right]\right\} \ , && \text{(C.37)}\\
b(t) &= \frac{\nu'^* e^{-\mathscr{C}t}}{\bar{n}(1-e^{-t}) + \epsilon e^{-\mathscr{C}t}} \ , && \text{(C.38)}\\
c(t) &= \frac{\nu' e^{-\mathscr{C}t}}{\bar{n}(1-e^{-t}) + \epsilon e^{-\mathscr{C}t}} \ , && \text{(C.39)}\\
d(t) &= \frac{1}{\bar{n}(1-e^{-t}) + \epsilon e^{-\mathscr{C}t}} \ . && \text{(C.40)}
\end{aligned}
$$

A substitution of these equations into Eq. (C.27) results in the Gaussian form

$$\phi_s(\nu,t) = \frac{1}{\pi m(t)} \exp\left[-\frac{|\nu - G(t)\nu'|^2}{m(t)}\right] , \tag{C.41}$$

which we recognize as a displaced spontaneous emission field (Eq. (C.7)). We here have the relations

$$G(t) = e^{-\mathscr{C}t} , \tag{C.42}$$

and

$$m(t) = \bar{n}(1 - e^{-\mathscr{C}t}) = -\bar{n}(|G(t)^2 - 1) . \tag{C.43}$$

Since $\phi_s(\nu,t)$ is a solution to the Fokker-Planck equation, Eq. (C.24),

$$\phi(\nu,t) = \int \phi_0(\nu')\phi_s(\nu - G(t)\nu', t)d^2\nu , \tag{C.44}$$

where $\phi_0(\nu')$ denotes the phase space density at zero-time, is also a solution. Hence we have shown that the amplified field may be regarded as a simple convolution of the input, or zero-time phase space density, with the phase space density associated with spontaneous emission.

Since we have $m(t) \to 0$ as $t \to 0$, we have

$$\lim_{t\to 0} \phi_s(\nu,t) = \begin{cases} \infty & \nu = 0 , \\ 0 & \nu \neq 0 , \end{cases} \tag{C.45}$$

which correspond to a δ-function, i.e.

$$\lim_{t\to 0} \phi_s(\nu - \nu', t) = \delta^2(\nu - \nu') . \tag{C.46}$$

We may therefore calculate

$$\begin{aligned} \phi(\nu,0) &= \lim_{t\to 0} \phi(\nu,t) \\ &= \lim_{t\to 0} \int \phi_0(\nu')\phi_s(\nu - G(t)\nu', t)d^2\nu \\ &= \int \phi_0(\nu')\delta^2(\nu - \nu')d^2\nu' \\ &= \phi_0(\nu) , \end{aligned} \tag{C.47}$$

so the initial condition is satisfied for $\phi(\nu,t)$.

Appendix D

Further Reading

- Center for Quantum Computing, www.qubit.org
- arXiv.org e-Print archive, http://arxiv.org
- M.O. Scully and M.S. Zubairy *"Quantum Optics"* (Cambridge University Press, 1997)
- S. Whealton, *"The Number Wars"*, http://washingtonart.net/whealton/wars.html
- D. Deutsch, *"Quantum Computation"*, Physics World (June 1992)
- H. Jeong and M.S. Kim, *"Efficient Quantum Computation Using Coherent States"*, quant-ph/019077
- C. Zalka, *"An Introduction to Quantum Computers"*, quant-ph/9811006
- G. Brassard, *"New Trends in Quantum Computing"*, quant-ph/9602014
- A. Steane, *"Quantum Computing"*, Rept. Prog. Phys. 61 117-173 (1998)
- H. Jeong and M.S. Kim, *"Entanglement Purification for Entagled Coherent States"*, quant-ph/0111015
- M.G.A. Paris, *"Optical Qubit Using Linear Elements"*, quant-ph/0009095
- S. Lloyd, *"Quantum Computation over Continuous Variables"*, quant-ph/9810082

☐ C. Adami and N.J. Cerf, *"Quantum Computation with Linear Optics"*, quant-ph/9806048

☐ N.J. Cerf, T. Durt and N. Gisin, *"Cloning a Qutrit"*, quant-ph/01100092

☐ S.D. Bartlett and B.C. Sanders, *"Universal Continuous-variable Quantum Computation: Requirement of Optical Nonlinearity for Photon Counting"*, quant-ph/0110039

☐ I.L. Chuang and Y. Yamamoto, *"Simple Quantum Computer"*, Phys. Rev. A **52**(5) 3489 (November 1995)

☐ E. Knill, R. Laflamme and G.J. Milburn, *"A Scheme for Efficient Quantum Computation with Linear Optics"*, Nature **409** (January 2001)

☐ G.J. Milburn, *"Quantum Optical Fredkin Gate"*, Phys. Rev. Lett. **62**(18) 2124-2127 (May 1989)

☐ S.P. Hotaling and A.R. Pirich (editors), *"Photonic Quantum Computing"*, Proceedings of SPIE **3076** (April 1997)

☐ T.T. Dao and D.M. Campbell, *"Multiple-valued Logic; an Implementation"*, Opt. Eng. **25**(1) 14-21 (January 1986)

☐ B.L. Drake et al., *"Photonic Computing Using the Modified Signed-digit Number Representation"*, Opt. Eng. **25**(1) 38-43 (January 1986)

☐ S.L. Hurst, *"Multiple-valued Threshold Logic; its Status and its Realization"*, Opt. Eng. **25**(1) 44-55 (January 1986)

☐ Ye Olde Thinkquest, http://library.thinkquest.org/19488/

Bibliography

[1] L. Mandel and E. Wolf, *"Optical Coherence and Quantum Optics"* (Cambridge University Press, 1995)

[2] D.F. Walls and G.J. Milburn, *"Quantum Optics"* (Springer, 1995)

[3] M.M. Mano *"Digital Design"* (Prentice-Hall International, 1991)

[4] M.A. Nielsen and I.L. Chuang, *"Quantum Computation and Quantum Information"* (Cambridge University Press, 2000)

[5] D.C. Rine, *"Computer Science and Multiple-valued Logic"* (North-Holland Publishing Company, 1977)

[6] A.A.S. Awwal and K.M. Iftekharuddin, *"Graphical Approach for Multiple-valued Logic Minimization"*, Opt. Eng. **38**(3) 462-467 (March 1999)

[7] Brian Hayes, *"Third Base"*, Am. Sci. **89**(6) 490 (November-December 2001)

[8] T. Hozumi, O. Kakusho and Y. Hata, *"On Low Cost Realization of Multiple-valued Logic Functions"*, Proc. 28th IEEE Int. Symp. on Multiple-valued Logic 233-238 (1998)

[9] N.P. Brousentsov, S.P. Maslov, J. Ramil Alvarez and E.A. Zhogolev, *"Development of Ternary Computers at Moscow State University"*, Russian Virtual Computer Museum, 1997-2002, http://www.computer-museum.ru/english/setun.htm