

## **Reversible and Endoreversible Computing**

**Alexis De Vos<sup>1</sup>**

*Received January 30, 1995*

---

Reversible combinatorial computers are built from basic cells with a three-bit digital input and a three-bit digital output. Such a computer can calculate both 'from left to right' and 'from right to left', such that input pins and output pins are indistinguishable. In order to perform a calculation in a specific direction, an electric field should be applied externally. The inevitably frictional losses occur in the lines supplying the computer with the input data and in the lines draining the calculation results to the output registers. Such behavior is analogous to the endoreversible operation of heat engines and other energy converters.

---

### **1. INTRODUCTION**

There exists a vast literature on the physics of computing. For review, see, e.g., Mead and Conway (1980) and Bennett and Landauer (1985). In this context, the need for a comparison between a computer and a Carnot heat engine has been mentioned several times (Maddox, 1987; Costa de Beauregard, 1989; Jablonski, 1990). However, we still lack a formal and rigorous theory of reversible computation (Wolpert, 1992). True, there exists a useful literature on logically reversible and irreversible gates (Bennett and Landauer, 1985; Fredkin and Toffoli, 1982). But the relationship between logical and physical reversibility is still not clear. In the present paper, a theoretical framework is presented in order to find a relationship between logical operations and physical quantities in a combinatorial computer. The tool which is proposed for this goal is endoreversible thermodynamics.

The general endoreversible engine consists of four reservoirs. The temperature  $T$  of each of them is constant, and so are their intensive quantities  $X, Y, \dots$ . Each reservoir can supply an energy current  $U$  as well as extensive quantity currents  $x, y, \dots$ , where each current  $x$  is associated with the corresponding parameter  $X$ , in such a way that the product  $xX$  is a work current.

<sup>1</sup>Vakgroep Elektronika en Informatiesystemen, Universiteit Gent, B-9000 Gent, Belgium.

The reservoirs 1 and 2 are the external reservoirs, whereas the reservoirs 3 and 4 are the internal or intermediate reservoirs. (See Fig. 1a.)

Between reservoir 1 and reservoir 3 is a conductor, between reservoir 3 and reservoir 4 is a reversible engine, and between reservoir 4 and reservoir 2 is a second conductor. The fact that the inner engine is reversible means that it satisfies the following two axioms:

*Axiom 1.* Conservation of energy:

$$U_1 - U_2 = W$$

*Axiom 2.* Conservation of entropy:

$$S_1 - S_2 = 0$$

where  $W$  denotes the produced power, i.e., work per unit time, and  $S$  is the entropy current associated with the energy current  $U$  and the currents  $x, y, \dots$ :

$$S = \frac{U - (xX + yY + \dots)}{T}$$

The subscript 1 (in  $U_1$  and  $S_1$ ) refers to currents supplied to the reversible core, whereas subscript 2 (in  $U_2$  and  $S_2$ ) refers to currents supplied by it (see

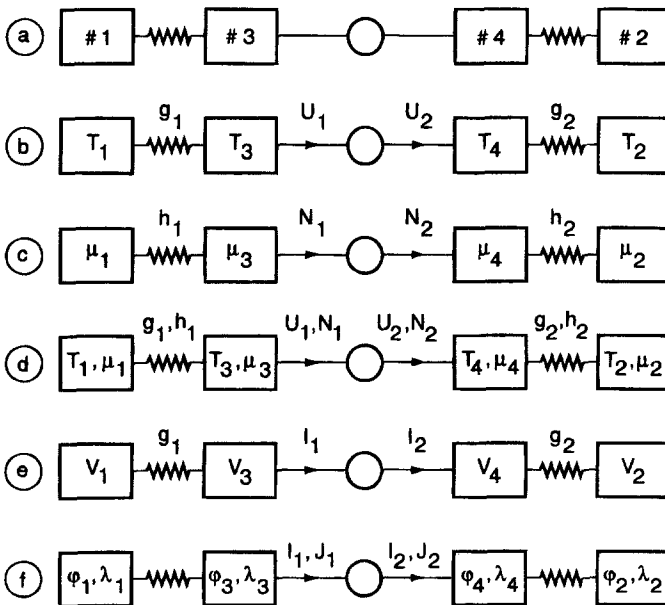


Fig. 1. Endoreversible engines: (a) general engine, (b) thermal engine, (c) chemical engine, (d) photovoltaic engine, (e) electrical network, (f) computational engine.

Figs. 1b to 1f). The condition  $S_1 - S_2 = 0$  expresses that no entropy is generated in the engine itself. This explains the name *endoreversible*: reversible in its internal parts (Rubin, 1979). Thus all entropy generation takes place in the two conductors, i.e., in the communication between the engine and the external world, the latter represented by reservoirs 1 and 2.

Figures 1b and 1c show two simple examples of endoreversible engines: the endoreversible thermal engine, due to Curzon and Ahlborn (1975), and the endoreversible chemical reactor, due to De Vos (1991a). In the former case, we have the following components:

- The reservoirs are heat reservoirs, i.e., reservoirs only labeled by their temperature  $T$  and providing only a heat current  $U$ .
- The conductors are assumed to obey simple Fourier–Newton transport equations:

$$U_1 = g_1(T_1 - T_3)$$

$$U_2 = g_2(T_4 - T_2)$$

where the coefficients  $g_1$  and  $g_2$  are thermal conductances.

- The reversible engine is an ideal Carnot engine.

In the latter case, we have the following components:

- The reservoirs are particle reservoirs, i.e., reservoirs only labeled by their chemical potential  $\mu$  and providing only a particle current  $N$ .
- The conductors are assumed to obey simple Fick laws:

$$N_1 = h_1 \left[ \exp\left(\frac{\mu_1}{kT}\right) - \exp\left(\frac{\mu_3}{kT}\right) \right]$$

$$N_2 = h_2 \left[ \exp\left(\frac{\mu_4}{kT}\right) - \exp\left(\frac{\mu_2}{kT}\right) \right]$$

where the coefficients  $h_1$  and  $h_2$  contain both diffusion coefficients and geometrical factors.

- The reversible engine is an ideal fuel cell.

Figure 1d shows a less simple example: the endoreversible photovoltaic engine (De Vos, 1991b). Here reservoirs need at least two labels: temperature  $T$  and chemical potential  $\mu$ . Between reservoirs, two extensive quantities are exchanged, i.e., energy  $U$  and particles  $N$ . The transport rates are governed by laws of the form

$$U_1 = g_1(T_1, \mu_1) - g_1(T_3, \mu_3)$$

$$U_2 = g_2(T_4, \mu_4) - g_2(T_2, \mu_2)$$

$$N_1 = h_1(T_1, \mu_1) - h_1(T_3, \mu_3)$$

$$N_2 = h_2(T_4, \mu_4) - h_2(T_2, \mu_2)$$

where  $g_1$ ,  $g_2$ ,  $h_1$ , and  $h_2$  are some mathematical functions: linear, exponential, or whatsoever.

Endoreversible models have proven their value, not only for thermal engines and chemical reactors, but also for describing solar energy conversion, climatology, economics, etc. A survey can be found in Sieniutycz and Salamon (1990) and De Vos (1992, 1993, 1995a).

## 2. ENDOREVERSIBLE LOGIC GATES

In order to relate computational gates to endoreversible engines, we start by introducing an electric engine. The electric network simply consists of two (ideal) voltage sources  $V_1$  and  $V_2$ , two (Ohmic) conductances  $g_1$  and  $g_2$ , and one (rechargeable) battery  $\Delta$  (see Fig. 2a). Its endoreversible schematics is given in Fig. 1e. The electric currents  $I_1$  and  $I_2$ , as a function of the battery voltage  $\Delta$ , are depicted in Fig. 2b. Finally, the power  $W$  stored by unit of time into the battery is displayed in Fig. 2c. According to the classical scheme of endoreversible thermodynamics, we distinguish three special values of the parameter  $\Delta$ :

- For  $\Delta = 0$ , we have a short-circuit current  $g(V_1 - V_2)$ , where  $g$  is short-hand notation for  $g_1 g_2 / (g_1 + g_2)$ .
- For  $\Delta = \frac{1}{2}(V_1 - V_2)$ , we have maximum power transfer  $W_{\max} = \frac{1}{4}g(V_1 - V_2)^2$ .
- For  $\Delta = V_1 - V_2$ , we have open circuit, i.e.,  $I_1 = I_2 = 0$ .

Now suppose we have dual power supplies  $\pm\varphi_1$  and  $\pm\varphi_2$  (both  $\varphi_1$  and  $\varphi_2$  denoting positive voltage levels). We have two one-bit registers:  $A$  and  $P$ . This means that register  $A$  is at voltage  $V_1$ , equaling either  $+\varphi_1$  or  $-\varphi_1$ , and that register  $P$  is at  $V_2$ , equal either to  $+\varphi_2$  or  $-\varphi_2$ . We assume that a negative voltage is a logic zero and a positive voltage is a logic one. The logic state of register  $A$  is denoted  $\lambda_1$ ; the logic state of register  $P$  is denoted  $\lambda_2$ . Between the two registers is a logic gate. As an introductory example we chose the simplest gate imaginable, i.e., the follower. Table Ia displays its truth table. A truth table tells how a logic operator  $\Omega$  operates on a logic variable: we write  $\Omega(\lambda_1) = \lambda_2$ . In the case of the follower, we have  $\Omega(0) = 0$  and  $\Omega(1) = 1$ .

The physical implementation of  $P$  following  $A$  consists of an ideal (i.e., lossless) switch  $S_3$  and two ideal batteries. If  $V_3$  is positive, then the switch is up; if  $V_3$  is negative, the switch is down. In order to make the follower

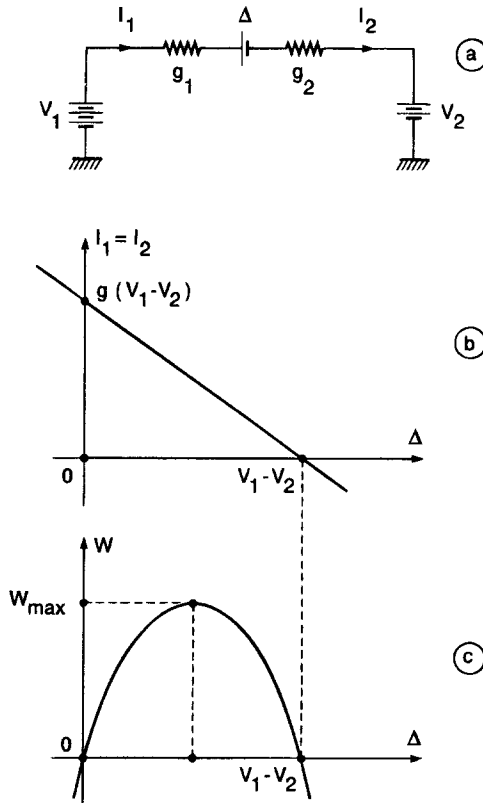


Fig. 2. Endoreversible electrical engine: (a) electrical network layout, (b) current–voltage characteristic, (c) work–voltage characteristic.

Table I. Truth Table of the Two Studied Gates

(a)		(b)					
A	P	A	B	C	P	Q	R
0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0
		0	1	0	1	0	1
		0	1	1	1	0	0
		1	0	0	0	1	1
		1	0	1	0	1	0
		1	1	0	0	0	1
		1	1	1	1	1	1

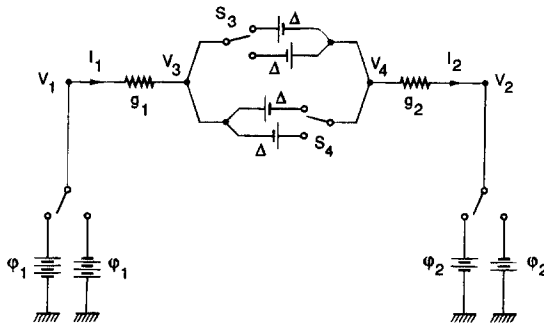


Fig. 3. Implementation of the one-bit follower.

physically reversible, also a switch  $S_4$  and two batteries are added to make  $A$  follow  $P$ . Switch  $S_4$  is up if voltage  $V_4$  is positive; it is down if  $V_4$  is negative. (See Fig. 3.) Note that again we have two lossy components, i.e., the finite electric conductances  $g_1$  and  $g_2$ .

We first consider a case where the logic equation  $\Omega(\lambda_1) = \lambda_2$  is fulfilled, i.e.,  $A = P = 1$ . Then, it is clear that the electric currents  $I_1 = I_2 = g(V_1 - V_2 - \Delta)$  equal  $g(\varphi_1 - \varphi_2 - \Delta)$ . (See Fig. 4.)

We show in Fig. 5 the voltage profile along the wire from  $A$  to  $P$ , in four different cases:

- $A = 0$  and  $P = 0$
- $A = 0$  and  $P = 1$
- $A = 1$  and  $P = 0$
- $A = 1$  and  $P = 1$

and two different subcases

- $\Delta < \varphi_1 - \varphi_2$
- $\Delta > \varphi_1 - \varphi_2$

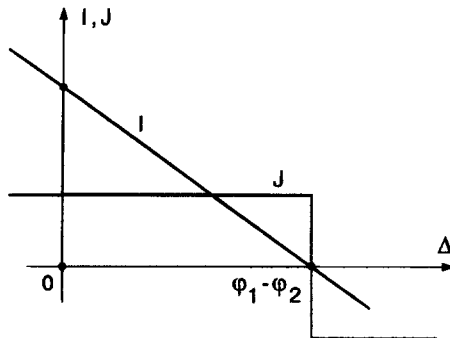


Fig. 4. Endoreversible behavior of the one-bit follower, for the case  $A = P = 1$ .

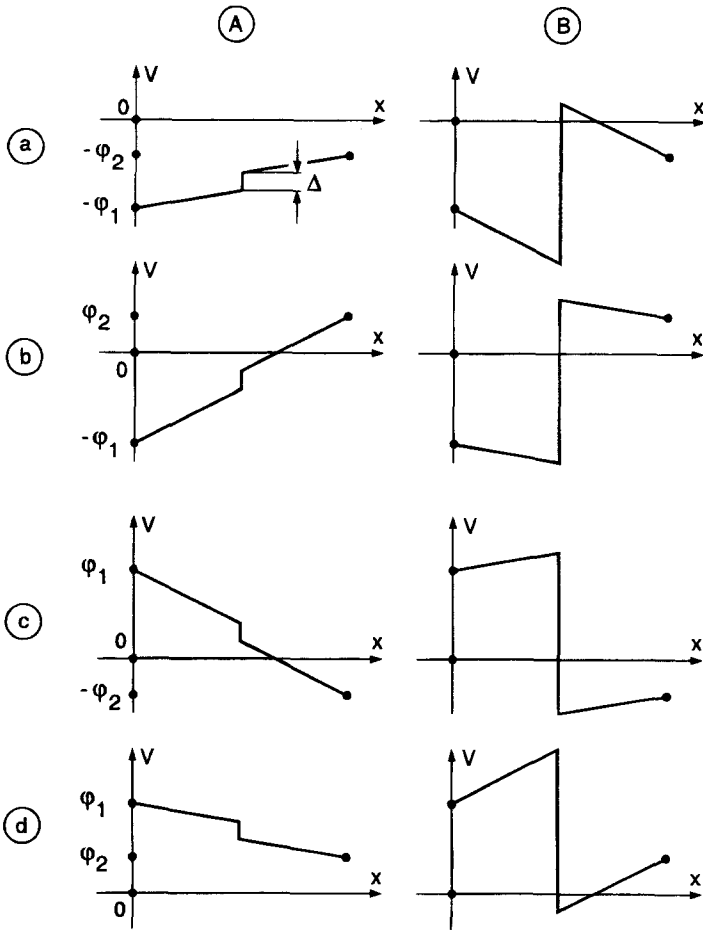


Fig. 5. Voltage profile in the one-bit follower.

Always the current is of the form  $g(\pm\varphi_1 \pm \varphi_2 \pm \Delta)$ .

We have until now considered registers as voltage and information sources. If the voltage is negative, a logic 0 is ‘sent’; if the voltage is positive, a logic 1 is ‘emitted’. Now we look at a register as a ‘receiver’ of logic information: if a negative current enters the register, a logic 0 is ‘read’; if a positive current enters the register, a logic 1 is ‘received’. Thus, in our circuit, register A reads the content of register P by measuring the current  $-I_1$ , whereas register P measures the current  $I_2$  in order to probe the logic content of A.

It is clear from the voltage gradients in Fig. 5 that:

- For  $\Delta < \varphi_1 - \varphi_2$ , register P receives correct information about the content of A whenever  $\Omega(\lambda_1) = \lambda_2$  is fulfilled (see Figs. 5Aa and 5Ad).

- For  $\Delta > \varphi_1 - \varphi_2$ , register  $A$  receives correct information about the content of  $P$  whenever  $\Omega(\lambda_1) = \lambda_2$  is fulfilled (see Figs. 5Ba and 5Bd).

In other words:

- For  $\Delta < \varphi_1 - \varphi_2$ , information flows from  $A$  to  $P$ .
- For  $\Delta > \varphi_1 - \varphi_2$ , information flows from  $P$  to  $A$ .

This is visualized in Fig. 4, where a positive  $J$  denotes information flow from  $A$  to  $P$  and negative  $J$  denotes information flow from  $P$  to  $A$ . Note that, for  $\Delta = \varphi_1 - \varphi_2$ , no information is transported.

We see in Fig. 4 that the point  $\Delta = \varphi_1 - \varphi_2$  is the reversible point: both the electric current  $I$  (short-hand notation for  $I_1 = I_2$ ) and the information current  $J$  are zero. In Fig. 4, the curve  $J(\Delta)$  is not continuous. This is caused by the fact that we assume noiseless channels (Shannon, 1948). In Appendix A we give an example of the behavior of noisy channels.

Thus a logic gate, functioning between two registers  $A$  and  $P$  with given content, operates reversibly if and only if it fulfills two conditions:

- Its batteries are tuned to  $\Delta = \varphi_1 - \varphi_2$ .
- Its truth table is satisfied.

These two conditions can be written in the same form:

$$\begin{aligned} (-\Delta + \varphi_1) - \varphi_2 &= 0 \\ \Omega(\lambda_1) - \lambda_2 &= 0 \end{aligned}$$

Note that in the second line the ‘minus sign’ between the two logic variables stands for the XOR operator (exclusive OR).

We can conclude that the continuous variable  $\Delta$  and the discrete variable  $\Omega$  of the logic gate play the same role as the two continuous variables  $T_3 - T_4$  and  $\mu_3 - \mu_4$  of the photovoltaic converter. The continuous variable  $\varphi$  and the discrete variable  $\lambda$  of the ‘data reservoirs’ of the computer (Fig. 1f) play the same role as the continuous labels  $T$  and  $\mu$  of the photovoltaic reservoirs (Fig. 1d). Compared to the general endoreversible engine (Fig. 1a), we may say that the role of the intensive variables  $X$  and  $Y$  is played by  $\varphi$  and  $\lambda$ , whereas the role of the corresponding extensive variables  $x$  and  $y$  is played by the electric current  $I$  and the information stream  $J$ . See Fig. 1f.

However, in practice, there is a significant difference in approach between the computational engine and the photovoltaic engine. Indeed, in the case of a photovoltaic engine, often the external properties are fixed. For example, in a solar energy converter  $T_1$  and  $\mu_1$  are related to the properties of the sun (e.g.,  $T_1 = 6000$  K and  $\mu_1 = 0$ ), whereas  $T_2$  and  $\mu_2$  are related to the earthly environment (e.g.,  $T_2 = 300$  K and  $\mu_2 = 0$ ). By changing the converter parameters  $T_3 - T_4$  and  $\mu_3 - \mu_4$  we influence the conversion



properties (e.g., the conversion efficiency). In the case of a calculator, usually the internal parameters  $\Delta$  and  $\Omega$  are fixed by the hardware. The external properties, i.e., the binary inputs  $\lambda_1 = (A, B, C, \dots)$  and  $\lambda_2 = (P, Q, R, \dots)$ , on the contrary, change frequently. Therefore we will assume  $\Delta$  and  $\Omega$  constant.

We will, e.g., choose  $\Delta = 0$  and  $\Omega$  given by the truth table of Table Ib. Why exactly this example? Because this logic gate with three inputs and three outputs plays a fundamental role as a basic building block for reversible Boolean computers. In Appendix B we demonstrate how it fulfills this role.

The reader will easily verify that all conclusions for the one-bit follower are equally valid for this three-bit reversible gate, and therefore for any reversible combinatorial network. One has only to generalize the meaning of the minus sign in the reversibility condition  $\Omega(\lambda_1) - \lambda_2 = 0$ . It now, e.g., stands for the Hamming distance, i.e., the number of ones in the logic variable  $(A \text{ XOR } P, B \text{ XOR } Q, C \text{ XOR } R, \dots)$ .

For sake of completeness, we remark that applying higher voltage sources at one side and lower (or zero) voltage sources at the other side of a logic network in order to measure output currents at the second side is not the usual way to perform electronic computing. The common practice is to apply voltage sources at the 'input' side and measure open-circuit voltages at the 'output' side. If we have described above a computer where output currents are measured, it is with the only purpose to illustrate better the analogy with the endoreversible heat engines.

Finally, we remark that in the above description we used linear (i.e., Ohmic) series resistances. In practice, the lossy components in series with the ideal switches are formed by the finite conductances of the MOS channels. These, of course, behave far from linearly. However, this fact does not influence the basic results of our endoreversible picture.

### 3. CONCLUSION

In the present paper we have discussed logically reversible gates. We have made a rational choice for a basic building block. We have modeled this building unit in the same way as an endoreversible heat engine.

We have demonstrated how the endoreversible computer works. In reversible conditions it dissipates no heat, but is useless, as it works at the edge of forward and backward operation. The computer hesitates between calculating in forward and in backward direction. This is completely similar to a reversible heat engine: in reversible conditions, the efficiency of a Carnot engine is maximal (i.e., equal to the Carnot efficiency), but this operation is useless, as the conversion of heat into work happens infinitely slowly: the engine hesitates between operation as heat engine and operation as heat pump.

In order to perform a really complete calculation (i.e., to take data in, to process the data, and to take resulting data out), it is necessary to operate the computer out of physical equilibrium, by introducing an external force, e.g., an electric field. Then necessarily some electrochemical energy is converted into heat in the bit conductors. In this endoreversible scheme, no energy is dissipated in the gates themselves of the combinatorial circuit: energy is only dissipated in the connections between registers and gates and in the interconnections between the various gates. These results are in agreement with the points of view developed by Feynman (1985).

## APPENDIX A. CHANNELS WITH NOISE

Noise introduces errors, such that, on the average, only a fraction  $\alpha$  of the bits emitted by one reservoir is received correctly by a second reservoir. Note that  $0 \leq \alpha \leq 1$ . We assume the voltage level  $\varphi_i$  larger than the voltage level  $\varphi_j$ , i.e. the situation where information transmission from  $i$  to  $j$  is aimed at (see Fig. 6a). What is then the rate  $J$  of transmission of information? Following Shannon (1948), who describes the special case  $\alpha = 99/100$ , we write

$$J = H(i) - H_j(i) \quad (\text{A1})$$

where  $H(i)$  is the entropy of the message sent by reservoir  $i$ , whereas  $H_j(i)$  is the conditional entropy. Let  $p_0$  and  $p_1 = 1 - p_0$  be the probabilities of register  $i$  emitting a 0 ( $V_i = -\varphi_i$ ) and a 1 ( $V_i = +\varphi_i$ ), respectively. If we assume  $p_0$  and  $p_1$  equal to  $1/2$ , then

$$\begin{aligned} H(i) &= -p_0 \log(p_0) - p_1 \log(p_1) \\ &= \log(2) \\ &= 1 \text{ bit} \end{aligned} \quad (\text{A2})$$

and

$$\begin{aligned} H_j(i) &= -\alpha p_0 \log \left[ \frac{\alpha p_0}{\alpha p_0 + (1 - \alpha) p_1} \right] - (1 - \alpha) p_0 \log \left[ \frac{(1 - \alpha) p_0}{(1 - \alpha) p_0 + \alpha p_1} \right] \\ &\quad - \alpha p_1 \log \left[ \frac{\alpha p_1}{\alpha p_1 + (1 - \alpha) p_0} \right] - (1 - \alpha) p_1 \log \left[ \frac{(1 - \alpha) p_1}{(1 - \alpha) p_1 + \alpha p_0} \right] \\ &= -\alpha \log(\alpha) - (1 - \alpha) \log(1 - \alpha) \end{aligned} \quad (\text{A3})$$

We first consider the case where the two logic variables are equal:  $\lambda_i = \lambda_j$ . Then the voltage difference  $V_i - V_j$  between the two terminals can be equal to one of the following two values: either  $-\varphi_i + \varphi_j$  or  $\varphi_i - \varphi_j$ .

If, e.g.,  $V_i = \varphi_i$  (i.e., logic 1) and  $V_j = \varphi_j$  (i.e., also logic 1), then the current  $I$  is supposed to be  $g(\varphi_i - \varphi_j - \Delta)$ . Assuming, e.g., Boltzmann statistics for crossing the energy barrier  $q \text{ abs}(\varphi_i - \varphi_j - \Delta)$ , there is a finite probability for the current to flow in the ‘wrong direction’, i.e., from low potential to high potential. The probability to have a correct current sign is thus lower than unity:

$$\alpha = \frac{1}{1 + \exp[-q \text{ abs}(\varphi_i - \varphi_j - \Delta)/kT]} \tag{A4}$$

Note that this simple model leads to valuable results: for  $\text{abs}(\varphi_i - \varphi_j - \Delta) \gg kT$  transmission is reliable ( $\alpha \approx 1$ ), whereas for  $\text{abs}(\varphi_i - \varphi_j - \Delta) \ll kT$  transmission is poor ( $\alpha \approx 1/2$ , such that the received bit is completely randomized). Substitution of (A4) into (A3) and subsequent substitution of (A2), (A3) into (A1) finally leads to a  $J$  curve as in Fig. 6b. This continuous curve replaces the discontinuous curve in Fig. 4, which is recovered in the limiting case of  $T \rightarrow 0$ .

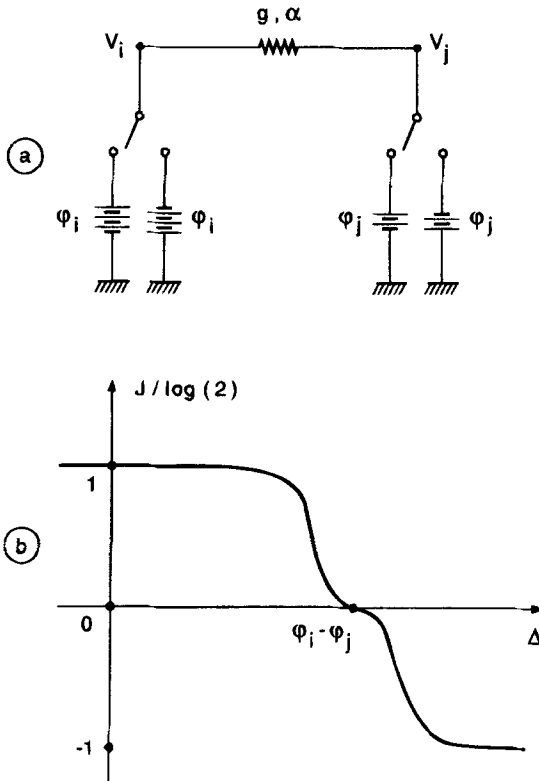


Fig. 6. Transmission rate of channel with noise.

We now consider the case where  $\lambda_i - \lambda_j$  is not zero. Then the energy barrier is  $q \text{ abs}(\varphi_i + \varphi_j - \Delta)$  and thus

$$\alpha = \frac{1}{1 + \exp[-q \text{ abs}(\varphi_i + \varphi_j - \Delta)/kT]}$$

Since usually  $\Delta = 0$  and  $\varphi_i + \varphi_j$  is large compared to  $\varphi_i - \varphi_j$ , this second case is less prone to errors due to noise.

## APPENDIX B. TEN LITTLE...

It has been demonstrated by Fredkin and Toffoli (1982) that in order to build a reversible layout of an arbitrary Boolean function, one needs gates with an equal number of inputs and outputs, and that this number should at least be equal to three. Well, there exist  $(2^3)^{(2^3)} = 8^8 = 16,777,216$  different truth tables with a three-bit input  $\lambda_1 = (A, B, C)$  and a three-bit output  $\lambda_2 = (P, Q, R)$ . Among these,  $(2^3)! = 8! = 40,320$  different tables are reversible truth tables. Table II shows four examples discussed in the literature:

- The notorious Fredkin and Toffoli (1982) gate.
- The gate that has been called the ‘controlled controlled not’ gate by Feynman (1985), but which Margolus (1988) calls the Toffoli gate.
- The gate called the ‘symmetric majority parity’ gate by Margolus (1988).
- The gate proposed by Peres (1985).

For convenience, we will call these gates the Fredkin gate, the Feynman gate, the Margolus gate, and the Peres gate, respectively. It is not clear at first sight what makes these four cases special among the many possibilities.

In order to make a rational choice among the 40,320 reversible gates with three inputs and three outputs, we first impose that the reverse gate should be identical to the gate itself. In other words: the gate should be its own reverse. This means the function  $A(P, Q, R)$  should be equal to the function  $P(A, B, C)$  and analogously for  $B$  and  $Q$  and for  $C$  and  $R$ . Such gates we will call symmetric gates. It has been shown elsewhere (De Vos, 1994a) that there exist 764 different symmetric gates. It is clear from Table II that the Fredkin gate and the Feynman gate are symmetric, but the Margolus gate and the Peres gate are not.

Next we require threefold rotational symmetry: internally  $A, B,$  and  $C$  should be indistinguishable and so should  $P, Q,$  and  $R$ . This means that the functions  $P(A, B, C), Q(B, C, A),$  and  $R(C, A, B)$  should be identical. Gates that have this threefold rotational symmetry we will call cyclic gates. Gates that have both the twofold mirror symmetry and the threefold rotational

symmetry, i.e., which are simultaneously symmetric and cyclic, we will call cyclosymmetric gates. The reader can easily verify there are only eight different cyclosymmetric gates. None of the gates of Table II is a member of this family.

Now we introduce a final condition: the gate should be able to operate without power supplies. This means the output signals  $P$ ,  $Q$ , and  $R$  should be deduced exclusively from the input signals  $A$ ,  $B$ , and  $C$ . No additional power inputs are allowed. This necessarily means that the first line of the truth table should read  $\Omega(000) = 000$  and the last line  $\Omega(111) = 111$ . Indeed, suppose we have an electronic computer where  $A$ ,  $B$ , and  $C$  are, e.g., at logic 0, thus at negative voltage. If power lines are missing, the gate is unable to fabricate any positive voltage in order to supply a logic one to either of the three outputs  $P$ ,  $Q$ , and  $R$ . The absence of power busbars and the subsequent conservation of 000 and 111 lowers the number of allowed gates from the eighth cyclosymmetric ones to only four. The remaining four truth tables are displayed in Table III. One of them, i.e., Table IIIa, is trivial, as it merely describes the three-bit follower, where  $P = A$ ,  $Q = B$ , and  $R = C$ . And then there were three.

So, we finally have to make a (rather arbitrary) choice among the remaining Tables IIIb–III d. We chose Table III d, as it has such simple rules of conversion:

- If  $A = B = C$ , then the outputs follow the inputs.
- Otherwise the outputs invert the inputs.

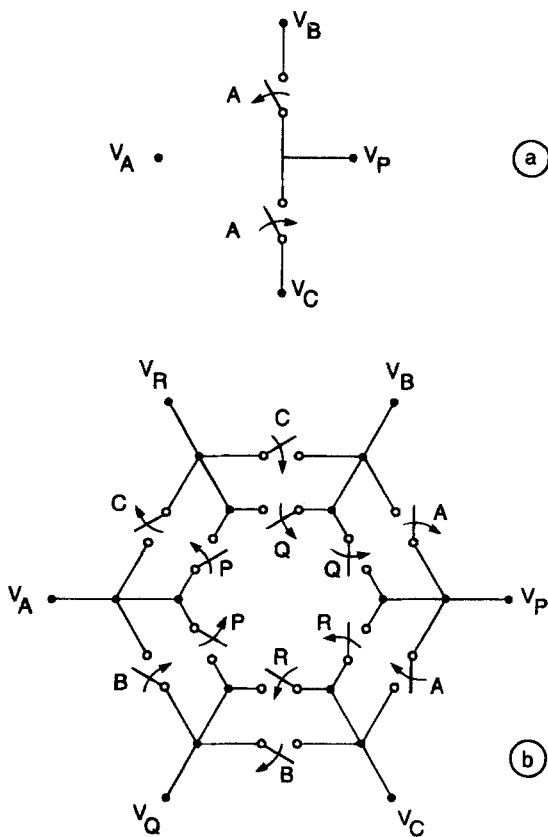
This final logic gate was first presented in De Vos (1994b) and its electronic implementation was described in De Vos (1995b). We recall here that its

**Table II.** Truth Table of Some Reversible Gates: (a) Fredkin Gate, (b) Feynman Gate, (c) Margolus Gate, (d) Peres Gate

(a)			(b)			(c)			(d)								
A	B	C	P	Q	R	A	B	C	P	Q	R	A	B	C	P	Q	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	
0	1	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	
0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	
1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	0	1	
1	1	0	1	1	0	1	1	0	1	1	1	1	0	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	

**Table III.** Truth Table of the Four Cyclosymmetric Reversible Gates Without Power Supply

(a)						(b)						(c)						(d)					
A	B	C	P	Q	R	A	B	C	P	Q	R	A	B	C	P	Q	R	A	B	C	P	Q	R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	1	
0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0	1	1	0	1	0	1	0	
0	1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	1	0	
1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	0	1	
1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	1	1	0	1	0	1	
1	1	0	1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	



**Fig. 7.** Construction of the basic logic gate: (a) inverting building block; (b) full gate consisting of six blocks.

following properties are important for implementing arbitrary Boolean functions:

- If one input is preset to logic zero, one output gives a logic OR function; e.g., if  $C = 0$ , then  $R = A \text{ OR } B$ .
- If one input is preset to logic one, one output gives a logic AND function; e.g., if  $C = 1$ , then  $R = A \text{ AND } B$ .
- If one input is preset to logic 0 and another one to logic 1, one output gives a logic NOT function; e.g., if  $B = 0$  and  $C = 1$ , then  $P = \text{NOT } A$ .

It is well known that the OR, AND, and NOT functions are sufficient for building any Boolean function. Therefore our logic gate is a universal primitive.

Its implementation for  $\Delta = 0$  is as follows. It consists of six ‘generalized inverters’ like Fig. 7a:

- If the voltages  $V_b$  and  $V_c$  have the same sign, then both switches are closed, such that the logic output  $P$  of the inverter equals  $B$  and equals  $C$ .
- If the voltages  $V_b$  and  $V_c$  have opposite sign, only one switch is closed, i.e., the one that realizes the logic inversion  $P = \text{NOT } A$ .

Three such ‘generalized inverters’ together form a ‘hexagon’ realizing the implementation  $(P, Q, R) = \Omega(A, B, C)$ . In order to realize the symmetric condition  $(A, B, C) = \Omega(P, Q, R)$ , a second ‘hexagon’ is added to the first one head to tail, leading to the complete gate in Fig. 7b. Also an implementation for arbitrary  $\Delta$  is possible. It is more complicated, as it contains not only additional batteries, but also additional switches. We will therefore not go into its details and limit ourselves to the zero- $\Delta$  case.

## ACKNOWLEDGMENTS

The Commission of the European Union is gratefully acknowledged for financial support through grant CIPA-CT 92-4026 in the framework of the Copernicus program.

## REFERENCES

- Bennett, C., and Landauer, R. (1985). The fundamental physical limits of computing, *Scientific American*, **253**(July), 38–46.
- Costa de Beauregard, O. (1989). The computer and the heat engine, *Foundations of Physics*, **19**, 725–732.
- Curzon, F., and Ahlborn, B. (1975). Efficiency of a Carnot engine at maximum power conditions, *American Journal of Physics*, **43**, 22–24.

- De Vos, A. (1991a). Endoreversible thermodynamics and chemical reactions, *Journal of Physical Chemistry*, **95**, 4534–4540.
- De Vos, A. (1991b). Is a solar cell an endoreversible engine? *Solar Cells*, **31**, 181–196.
- De Vos, A. (1992). *Endoreversible Thermodynamics of Solar Energy Conversion*, Oxford University Press, Oxford.
- De Vos, A. (1993). The endoreversible theory of solar energy conversion: A tutorial, *Solar Energy Materials and Solar Cells*, **31**, 75–93.
- De Vos, A. (1994a). Implementation of reversible logic gates in c-MOS, *International Journal of Electronics*, **76**, 293–302.
- De Vos, A. (1994b). Reversible computing in c-MOS, in *Proceedings of the Advanced Training Course on Mixed Design of VLSI Circuits*, Dębe, pp. 36–41.
- De Vos, A. (1995a). Thermodynamics of photochemical solar energy conversion, in *Proceedings of the 10th International Conference on Photochemical Conversion and Storage of Solar Energy*, Interlaken, 24–29 July 1994, *Solar Energy Materials and Solar Cells*, **38**, 11–22.
- De Vos, A. (1995b). A 12-transistor c-MOS building-block for reversible computers, *International Journal of Electronics*, **79**, 171–182.
- Feynman, R. (1985). Quantum mechanical computers, *Optics News*, **11**, 11–20.
- Fredkin, E., and Toffoli, T. (1982). Conservative logic, *International Journal of Theoretical Physics*, **21**, 219–253.
- Jablonski, D. (1990). A heat engine model of a reversible computation, *Proceedings of the IEEE*, **78**, 817–825.
- Maddox, J. (1987). Quantum information storage, *Nature*, **327**, 97.
- Margolus, N. (1988). Physics and computation, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, pp. 24–25.
- Mead, C., and Conway, L. (1980). *Introduction to VLSI Systems*, Addison-Wesley, Reading, Massachusetts, pp. 333–371.
- Peres, A. (1985). Reversible logic and quantum computers, *Physical Review A*, **32**, 3266–3276.
- Rubin, M. (1979). Optimal configuration of a class of irreversible engines, *Physical Review A*, **19**, 1272–1276.
- Shannon, C. (1948). The theory of communication, *Bell System Technical Journal*, **27**, 379–423.
- Sieniutycz, S., and Salamon, P. (eds.) (1990). *Finite-Time Thermodynamics and Thermoeconomics*, Taylor & Francis, New York.
- Wolpert, D. (1992). Reversible computing and physical laws, *Physics Today*, **45**(March), 98–99.