Generating the group of reversible logic gates

2002 J. Phys. A: Math. Gen. 35 7063

(http://iopscience.iop.org/0305-4470/35/33/307)

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS A: MATHEMATICAL AND GENERAL

J. Phys. A: Math. Gen. **35** (2002) 7063–7078 PII: S0305-4470(02)34943-6

Generating the group of reversible logic gates

Alexis De Vos[1], **Birger Raa**[2] **and Leo Storme**[3]

[1] Imec v.z.w., Vakgroep elektronika en informatiesystemen, Universiteit Gent, Sint Pietersnieuwstraat 41, B-9000 Gent, Belgium
[2] Vakgroep technische bedrijfskunde, Universiteit Gent, Technologiepark Zwijnaarde 9, B-9052 Gent, Belgium
[3] Vakgroep zuivere wiskunde en computeralgebra, Universiteit Gent, Galglaan 2, B-9000 Gent, Belgium

Received 15 March 2002, in final form 18 June 2002
Published 7 August 2002
Online at stacks.iop.org/JPhysA/35/7063

Abstract
Reversible logic plays a fundamental role both in ultra-low power electronics and in quantum computing. It is therefore important to have an insight into the structure of the group formed by the reversible logic gates and their cascading into reversible circuits. Such insight is gained from constructing chains of maximal subgroups. The subgroup of control gates plays a prominent role, as it is a Sylow 2-subgroup.

PACS numbers: 02.10.Ab, 02.20.−a, 03.67.Lx, 84.30.Bv

1. Introduction

Conventional computers are built from basic building blocks, such as the AND, NAND, OR, NOR and XOR gates. Such logic operations are logically irreversible. This means that, if we forget the value of the two inputs, knowledge of the output is not sufficient to calculate backwards and to recover the value of the inputs.

According to Landauer's principle [1–5], logic computations that are not reversible, necessarily generate heat, i.e. $kT\log(2)$, for every bit of information that is lost. Here k is Boltzmann's constant and T the temperature. For T equal to room temperature, this quantum of heat is small, i.e. 2.9×10^{-21} J, but non-negligible. In order to produce zero heat, a computer is only allowed to perform reversible computations. Such a logically reversible computation can be 'undone': the value of the output suffices to recover what the value of the input 'has been'. The hardware of such a reversible computer cannot be constructed from the conventional, i.e. irreversible gates. In contrast, it consists exclusively of logically reversible building blocks. The number of output columns of a reversible truth table necessarily equals its number of input columns. We call this number the 'logic width' of the gate.

The truth table of a logic gate of width w consists of 2^w lines, each containing two w-bit numbers: the w-bit input $(A_1, A_2, \ldots, A_w)$ and the w-bit output $(P_1, P_2, \ldots, P_w)$. For convenience, all possible inputs, ranging from $(0, 0, 0, \ldots, 0)$ to $(1, 1, 1, \ldots, 1)$, are ordered

A	P
0	1
1	0

(a)

A_1A_2	P_1P_2
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

(b)

$A_1\,A_2\,A_3$	$P_1\,P_2\,P_3$
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 0
0 1 1	0 1 1
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 1
1 1 1	1 1 0

(c)

Figure 1. Feynman's reversible truth tables: (*a*) NOT, (*b*) CONTROLLED NOT and (*c*) CONTROLLED CONTROLLED NOT.

arithmetically. Such a gate is reversible if and only if all 2^w output numbers form a permutation of the 2^w input numbers. Hence, there exist exactly $(2^w)!$ different reversible gates of width w. We now define the operation of cascading two gates (g_2 following g_1) of equal width: we simply bring the output P_i of the first gate g_1 to the corresponding input A_i of the second gate g_2, for each i satisfying $1 \leqslant i \leqslant w$. The set of reversible gates of width w, together with the operation of cascading, forms a group $\mathbf{R}_w$, isomorphic to the symmetric group $\mathbf{S}_{2^w}$ (of degree 2^w and order $(2^w)!$), the symmetric group $\mathbf{S}_n$ being defined as the group of *all* permutations of n elements. The cascading of two gates is not commutative; the cascading of more than two gates is associative. The number of gates cascaded, we call the depth of the computation.

Feynman [6, 7] introduced three particular reversible gates: the NOT, the CONTROLLED NOT and the CONTROLLED CONTROLLED NOT (see figure 1). These truth tables express the following boolean relations:

$$P = \texttt{NOT}\ A$$

for the NOT gate,

$$P_1 = A_1 \qquad P_2 = A_1\ \texttt{XOR}\ A_2$$

for the CONTROLLED NOT gate and

$$P_1 = A_1 \qquad P_2 = A_2 \qquad P_3 = (A_1\ \texttt{AND}\ A_2)\ \texttt{XOR}\ A_3$$

for the CONTROLLED CONTROLLED NOT gate. Because the output lines of the truth table of a reversible gate are just a permutation of the input lines, the truth table can be written in a condensed permutation notation. The NOT gate is represented by the permutation (1, 2) of the elements {1, 2}. The CONTROLLED NOT is the permutation (3, 4) of the elements {1, 2, 3, 4} and the CONTROLLED CONTROLLED NOT is the permutation (7, 8) of {1, 2, 3, 4, 5, 6, 7, 8}. The permutation of a cascade of two gates (g_2 following g_1) is the product $g_2 \cdot g_1$ of the two corresponding permutations.

Feynman's concept is extrapolated by Toffoli [8]. The gate, which Toffoli called the AND/NAND function of order w, was renamed later, in the framework of quantum computing

$A_1\ A_2\ A_3$	$P_1\ P_2\ P_3$
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 0
0 1 1	0 1 1
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 0
1 1 1	1 1 1

(a)

$A_1\ A_2\ A_3$	$P_1\ P_2\ P_3$
0 0 0	0 0 0
0 0 1	0 1 0
0 1 0	0 0 1
0 1 1	0 1 1
1 0 0	1 0 0
1 0 1	1 1 0
1 1 0	1 0 1
1 1 1	1 1 1

(b)

$A_1\ A_2\ A_3$	$P_1\ P_2\ P_3$
0 0 0	0 1 0
0 0 1	0 1 1
0 1 0	0 0 0
0 1 1	0 0 1
1 0 0	1 1 0
1 0 1	1 1 1
1 1 0	1 0 0
1 1 1	1 0 1

(c)

Figure 2. Three special truth tables within $\mathbf{R}_3$: (a) the follower, (b) an exchanger and (c) an inverter.

[9, 10], as the CONTROLLED^q NOT, a gate of width $w = q + 1$, satisfying

$$P_i = A_i \qquad \text{for all} \quad i \in \{1, 2, \ldots, w-1\}$$
$$P_w = (A_1 \ \texttt{AND}\ A_2 \ \texttt{AND} \cdots \texttt{AND}\ A_{w-1})\ \texttt{XOR}\ A_w.$$

Its permutation notation is $(2^w - 1, 2^w)$. A further generalization was introduced by Desoete and De Vos [11, 12], in the framework of ultra-low power electronics:

$$P_i = A_i \qquad \text{for all} \quad i \in \{1, 2, \ldots, w-1\}$$
$$P_w = f(A_1, A_2, \ldots, A_{w-1})\ \texttt{XOR}\ A_w$$

where f denotes an arbitrary boolean function of $q = w - 1$ boolean arguments. Such a gate is called a simple control gate. Now, there exist 2^{2^q} different boolean functions of q boolean variables. Together with the XOR operation, they form an Abelian (i.e. commutative) group $\mathbf{B}_q$, isomorphic to $\mathbf{Z}_2^{2^q}$, where $\mathbf{Z}_2$ is the cyclic group of order 2 (i.e. the group consisting of the two permutations ( ) and (1, 2) of the two elements {1, 2}). The set of simple control gates of width w, together with the operation of cascading, forms a group $\mathbf{c}_w$, isomorphic to $\mathbf{B}_{w-1}$. A simple control gate is represented by a permutation consisting of a product of disjoint transpositions of the form (2i − 1, 2i) with any integer i satisfying $1 \leqslant i \leqslant 2^{w-1}$. Therefore, the group of simple control gates can also be interpreted as the wreath product of $\mathbf{Z}_2$ with $\mathbf{1}_{w-1}$. Indeed, in such a product the elements $\{1, 2, 3, 4, \ldots, 2^w - 1, 2^w\}$ are partitioned into subsets of two elements, i.e. into $\{\{1, 2\}, \{3, 4\}, \ldots, \{2^w - 1, 2^w\}\}$, the group $\mathbf{Z}_2$ acting within the subsets and the group $\mathbf{1}_{w-1}$ acting between the subsets. Here, $\mathbf{1}_n$ represents the trivial group consisting of only the identity gate of width n (see figure 2(a) for the example $n = 3$).

Finally, the concept of simple control gates can be generalized towards control gates [11, 12]. A control gate of width w satisfies the relations:

$$P_i = f_i(A_1, A_2, \ldots, A_{i-1})\ \texttt{XOR}\ A_i \qquad \text{for all} \quad i \in \{1, 2, \ldots, w\}$$

where f_i is an arbitrary boolean function of $i - 1$ variables. Figure 3(a) shows a diagram of a control gate of width $w = 3$. Together with the operation of cascading, the control gates form a (non-Abelian) group $\mathbf{C}_w$ of order $2 \times 2^2 \times 2^3 \times \cdots \times 2^{2^{w-2}} \times 2^{2^{w-1}} = 2^{2^w - 1}$, isomorphic to the semidirect product $\mathbf{B}_{w-1} : \mathbf{B}_{w-2} : \cdots : \mathbf{B}_2 : \mathbf{B}_1$. In the next section, we will see that the order of $\mathbf{C}_w$, i.e. $2^{2^w - 1}$, reveals that $\mathbf{C}_w$ is nothing else but one of the Sylow 2-subgroups of $\mathbf{S}_{2^w}$.

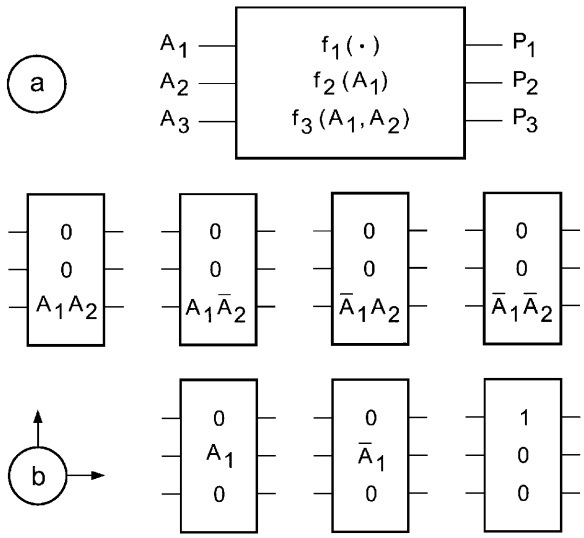

Figure 3. The control group $\mathbf{C}_3$: (*a*) arbitrary gate and (*b*) seven generating gates.

It is clear that simple control gates form a special case of control gates, with all $f_i = 0$ except f_w. As an example, the CONTROLLED CONTROLLED NOT is an element of the group $\mathbf{C}_3$ with $f_1 = 0$, $f_2(A_1) = 0$ and $f_3(A_1, A_2) = A_1$ AND A_2.

Control gates are attractive components, because they are easy to implement into hardware. Applying branch-based logic [13], i.e. basically wiring its control function as a sum of minterms, a simple control gate of width w needs an appropriate series-and-parallel connection of up to $(w-1)2^w$ switches, depending on the particular control function f [12]. Thus a control gate can be implemented by combining up to $(w-2)2^{w+1}+4$ switches, depending on the particular set of control functions $\{f_1, f_2, \ldots, f_w\}$. Applying Shannon decomposition can reduce the number of switches [11].

2. Sylow 2-subgroups of $\mathbf{S}_{2^n}$

We now remark that the order of $\mathbf{S}_{2^n}$ can be written as

$$(2^n)! = 2^{x(n)} \cdot y(n)$$

where we assume that $x(n)$ is the 'true' exponent of 2 in the factorization of $(2^n)!$, i.e. $y(n)$ is odd. The value of $x(n)$ can easily be calculated, as it equals the number of even factors in $1 \times 2 \times 3 \times 4 \times \cdots \times (2^n - 1) \times 2^n$, augmented with the number of quadruple factors in it, etc. Thus

$$x(n) = \frac{2^n}{2} + \frac{2^n}{4} + \cdots + 1 = 2^{n-1} + 2^{n-2} + \cdots + 1 = 2^n - 1.$$

Thus the number $2^{x(n)}$ equals the order of the subgroup $\mathbf{C}_n$. We may conclude that the group of control gates is one of the Sylow 2-subgroups of the group $\mathbf{R}_n$ of reversible gates. Indeed, Sylow's first theorem states that

> for every finite group and for every prime p which divides the order of the group, there exists at least one subgroup of order p^m, where m is the largest integer for which p^m divides the order of the group; such a subgroup is called a Sylow p-subgroup.

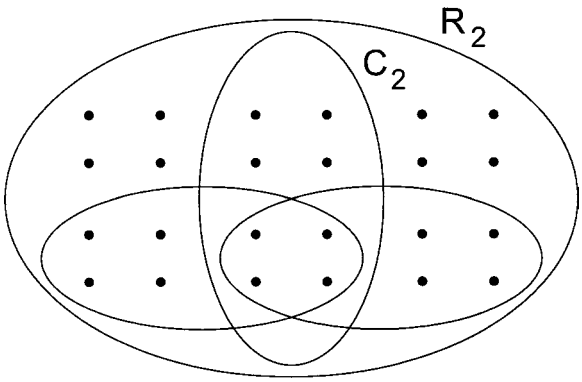

Figure 4. The three Sylow 2-subgroups of the symmetric group $\mathbf{S}_4$.

Sylow's second theorem states that

> the number of Sylow p-subgroups is congruent to 1 modulo p, i.e. a multiple of p plus one.

In our case ($p = 2$), this implies that the number of Sylow 2-subgroups is odd. Further, Sylow's third theorem states that

> the number of Sylow p-subgroups divides the order of the group.

In our case, this implies that the number of Sylow 2-subgroups divides $y(n)$. It equals the order of the parent group divided by the order of the normalizer of the subgroup. According to a theorem of Cárdenas and Lluis [14], the normalizer of a Sylow p-subgroup $\mathbf{G}_n$ of $\mathbf{S}_{p^n}$ equals $(\mathbf{Z}_{p-1})^n * \mathbf{G}_n$, where $*$ stands for split extension. For the case $p = 2$, we can conclude that the normalizer is the subgroup $\mathbf{G}_n$ itself. Hence, the number of Sylow 2-subgroups is $(2^n)!/2^{x(n)}$, i.e. $y(n)$ itself.

The value $x(w) = 2^w - 1$ thus explicitly shows the fact that $\mathbf{C}_w$ is a Sylow 2-subgroup of $\mathbf{R}_w$. As an example, figure 4 displays the case of $n = 2$, where $\mathbf{R}_2$ (isomorphic to $\mathbf{S}_4$) has $4! = 2^3 \cdot 3 = 24$ elements and $\mathbf{C}_2$ has $2^3 = 8$ elements. The other two Sylow 2-subgroups, conjugate to $\mathbf{C}_2$, are $(2, 3) \cdot \mathbf{C}_2 \cdot (2, 3)$ and $(1, 3) \cdot \mathbf{C}_2 \cdot (1, 3)$. This accords with the fourth theorem of Sylow stating that

> any two Sylow p-subgroups are conjugate.

For more details, the reader is referred to the existing literature on the Sylow p-subgroups of the symmetric group $\mathbf{S}_{p^n}$ with p an arbitrary prime number [15–19].

3. Generating the boolean group $\mathbf{B}_q$

Among the 2^{2^q} boolean functions of q arguments, there are 2^q functions having only one 1 in the output column of the truth table. See the functions $\varphi_1(A_1, A_2) = A_1 \,\mathtt{AND}\, A_2$, $\varphi_2(A_1, A_2) = A_1 \,\mathtt{AND}\, (\mathtt{NOT}\, A_2)$, $\varphi_4(A_1, A_2) = (\mathtt{NOT}\, A_1) \,\mathtt{AND}\, A_2$ and $\varphi_8(A_1, A_2) = (\mathtt{NOT}\, A_1) \,\mathtt{AND}\, (\mathtt{NOT}\, A_2)$ in table 1, for the case $q = 2$. Such functions we call minterm functions. In general, a minterm has the form

$$\tilde{A}_1 \,\mathtt{AND}\, \tilde{A}_2 \,\mathtt{AND} \cdots \mathtt{AND}\, \tilde{A}_n \tag{1}$$

where the notation $\tilde{A}_i$ stands for 'either A_i or $\mathtt{NOT}\, A_i$'.

Table 1. The 16 different functions φ of two boolean variables A_1 and A_2. The four columns are ordered according to the arithmetic order of the inputs (A_1, A_2), whereas the 16 rows are ordered according to the arithmetic order of the outputs $\varphi(A_1, A_2)$. Note that φ_1 is the AND function, φ_6 is the XOR function, φ_7 is the OR function, φ_8 is the NOR function and φ_{14} is the NAND function.

A_1A_2	0 0	0 1	1 0	1 1
φ_0	0	0	0	0
φ_1	0	0	0	1
φ_2	0	0	1	0
φ_3	0	0	1	1
φ_4	0	1	0	0
φ_5	0	1	0	1
φ_6	0	1	1	0
φ_7	0	1	1	1
φ_8	1	0	0	0
φ_9	1	0	0	1
φ_{10}	1	0	1	0
φ_{11}	1	0	1	1
φ_{12}	1	1	0	0
φ_{13}	1	1	0	1
φ_{14}	1	1	1	0
φ_{15}	1	1	1	1

The reader will easily verify that the minterm functions generate all elements of the group $\mathbf{B}_q$. In other words, each function f can be written as a XOR of minterm functions. For example, in table 1, we have

$$\varphi_7 = \varphi_1 \ \mathtt{XOR}\ \varphi_2 \ \mathtt{XOR}\ \varphi_4.$$

This property is a variant of the well-known theorem that any boolean function can be written as a sum (i.e. an OR) of minterms.

If we make an ordered set by taking the identity function φ_0 and subsequently adding the minterm functions $\varphi_1, \varphi_2, \varphi_4, \ldots, \varphi_{2^{2^q-1}}$, we generate a chain of subsequent subgroups of index 2. The first subgroup $\mathbf{1}$ is the trivial subgroup consisting only of the identity element; the following consists of the identity function and the AND function φ_1; ...; the last subgroup is the whole group $\mathbf{B}_q$:

$$\mathbf{1} = \{\varphi_0\} \subset \{\varphi_0, \varphi_1\} \subset \{\varphi_0, \varphi_1, \varphi_2, \varphi_3\} \subset \cdots \subset \{\varphi_0, \varphi_1, \ldots, \varphi_{2^{2^q}-2}, \varphi_{2^{2^q}-1}\} = \mathbf{B}_q.$$

There are $2^q + 1$ links in the chain. The chain is a chain of stabilizers and the elements $\varphi_1, \varphi_2, \varphi_4, \ldots$ are its strong generators [20]. This means the following: each finite group is isomorphic to some group of permutations of a set of elements (here $\mathbf{B}_q$ is isomorphic to a permutation group of 2^{q+1} elements); each stabilizer permutes more elements of the set than the previous stabilizer, i.e. the stabilizer to the left of it in the chain (equivalently, each stabilizer fixes more elements than the stabilizer to its right). We thus have not only the generators of $\mathbf{B}_q$, but also the generators for each link in the chain. Such strong generators are powerful tools for computational group theory [20].

Instead of the 2^q minterms, we can apply equally well the 2^q piterms of the Reed–Muller expansion [21]. These have also the form (1), but with another meaning for $\tilde{A}_i$: it stands for 'either A_i or 1'. In our example ($q = 2$), the piterms are φ_1, φ_3, φ_5 and φ_{15}, the Reed–Muller expansion of φ_7 now being

$$\varphi_7 = \varphi_1 \ \mathtt{XOR}\ \varphi_3 \ \mathtt{XOR}\ \varphi_5.$$

4. Generating the group of control gates

The minterm procedure of the previous section can, of course, be applied to generate the group $\mathbf{c}_w$ of simple control gates of width $w = q + 1$. In order to generate the group $\mathbf{C}_w$ of control gates, we choose generators as follows:

- First we take the control gate with all control functions f_i equal to zero, i.e. the identity gate.
- Then we take control gates with control functions $f_1, f_2, \ldots$ and f_{w-1} all equal to zero, but f_w equal to φ_{2^j} with j subsequently equal to $0, 1, 2, 3, \ldots$ and $2^q - 1$.
- Next we choose all control functions f_i equal to zero, except f_{w-1}, which is equal to φ_{2^j} with j subsequently equal to $0, 1, 2, 3, \ldots$ and $2^{q-1} - 1$.
- $\ldots$
- We choose all control functions f_i equal to zero, except f_2, which is equal to φ_{2^j} with j subsequently equal to 0 and 1.
- Finally we choose all control functions f_i equal to zero, except f_1, which is equal to $\varphi_{2^0} = \varphi_1 = 1$.

In this way we generate the whole group by stepwise enlarging the subgroup by an index equal to 2. The resulting chain consists of 2^w links:

$$\overbrace{\mathbf{1}_w \subset \mathbf{f}_w \subset \cdots \subset \underbrace{\mathbf{c}_w}}^{(2^{w-1}+1)\ \text{links}} \subset \cdots \subset \underbrace{\mathbf{C}_w}_{2^{w-1}\ \text{links}}.$$

Here $\mathbf{f}_w$ denotes the subgroup of order 2 consisting of the w-bit follower and the $\texttt{CONTROLLED}^{w-1}\,\texttt{NOT}$ gate. Figure 3(b) gives the diagrams of the seven generators of the group $\mathbf{C}_3$.

The reader can easily construct an analogous chain, applying a piterm procedure.

5. Generating the group of reversible gates

In order to enlarge $\mathbf{C}_w$ further to the whole group $\mathbf{R}_w$, we have to add generators which are not control gates. In order to find the groups between $\mathbf{C}_w$ and $\mathbf{R}_w$, we start from the $\mathbf{R}_w$ side, by looking for a maximal subgroup of it (a maximal subgroup $\mathbf{M}$ of a group $\mathbf{G}$ being defined by the fact that $\mathbf{M} \subset \mathbf{G}$ and there exist no subgroup $\mathbf{N}$ of $\mathbf{G}$ satisfying $\mathbf{M} \subset \mathbf{N} \subset \mathbf{G}$).

After a theorem independently given by O'Nan and Scott [22, 23], any maximal subgroup of a symmetric group $\mathbf{S}_n$ is either isomorphic to the alternating group $\mathbf{A}_n$ (defined as the group of all *even* permutations of n elements) or is a member of one of six special classes. Here we pick out one of these classes: maximal subgroups of $\mathbf{S}_{mk}$ of degree mk (with both $m > 1$ and $k > 1$) which are isomorphic to a wreath product of a symmetric group of degree m and a symmetric group of degree k. Applying the theorem for $m = 2^a$ and $k = 2^{w-a}$, with a an integer satisfying $1 \leqslant a \leqslant w - 1$, we find that there exist maximal subgroups of $\mathbf{R}_w$ isomorphic to the wreath product of $\mathbf{S}_{2^a}$ and $\mathbf{S}_{2^{w-a}}$. In order to construct such product, we partition the set of elements $\{1, 2, 3, \ldots, 2^w\}$ into subsets of 2^a subsequent elements, i.e. into $\{\{1, 2, \ldots, 2^a\}, \{2^a + 1, 2^a + 2, \ldots, 2^{a+1}\}, \ldots, \{2^w - 2^a + 1, 2^w - 2^a + 2, \ldots, 2^w\}\}$. The resulting wreath product $\mathbf{S}_{2^a}$ wr $\mathbf{S}_{2^{w-a}}$ is of the order

$$\begin{aligned}\text{order}\,(\mathbf{S}_{2^a}\ \text{wr}\ \mathbf{S}_{2^{w-a}}) &= [\text{order}\,(\mathbf{S}_{2^a})]^{\text{degree}\,(\mathbf{S}_{2^{w-a}})} \times \text{order}\,(\mathbf{S}_{2^{w-a}}) \\ &= [(2^a)!]^{2^{w-a}} \times (2^{w-a})!\end{aligned}$$

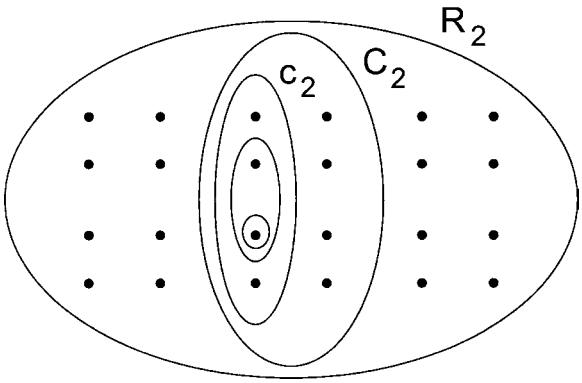

Figure 5. The chain of subgroups of the reversible group $\mathbf{R}_2$.

$$= [2^{x(a)} \cdot y(a)]^{2^{w-a}} \times 2^{x(w-a)} \cdot y(w-a)$$
$$= 2^{x(w)} \cdot [y(a)]^{2^{w-a}} \cdot y(w-a).$$

Thus the order of a maximal subgroup of $\mathbf{R}_w$ has the same true power of 2 as the order of $\mathbf{R}_w$ itself, but fewer odd prime factors.

If we choose $a = 1$, i.e. if we divide the set of elements into duos, we generate a subgroup isomorphic to $\mathbf{S}_2$ wr $\mathbf{S}_{2^{w-1}}$, of order $2^{x(w)} \cdot y(w-1)$. Next we can take duos of duos $\{\{\{1,2\},\{3,4\}\},\ldots\}$, leading us to the product $\mathbf{S}_2$ wr ($\mathbf{S}_2$ wr $\mathbf{S}_{2^{w-2}}$), of order $2^{x(w)} \cdot y(w-2)$. Proceeding further like this, we build a descending chain of subgroups of subsequent orders $2^{x(w)} \cdot y(w)$, $2^{x(w)} \cdot y(w-1)$, $2^{x(w)} \cdot y(w-2)$, ..., $2^{x(w)} \cdot y(2)$ and $2^{x(w)} \cdot y(1) = 2^{x(w)}$. The final subgroup, isomorphic to $\mathbf{S}_2$ wr ($\mathbf{S}_2$ wr ($\mathbf{S}_2$ wr ($\cdots$ ($\mathbf{S}_2$ wr $\mathbf{S}_2$) $\cdots$))), is nothing else but the control group, isomorphic to $\mathbf{Z}_2^{x(w)}$.

Together with the result of section 4, this finally yields a chain consisting of $2^w + w - 1$ links:

$$\overbrace{\mathbf{1}_w \subset \mathbf{f}_w \subset \cdots \subset \mathbf{c}_w}^{(2^{w-1}+1)\text{ links}} \underbrace{\subset \cdots \subset \overbrace{\mathbf{C}_w}^{w\text{ links}} \subset \cdots \subset \mathbf{R}_w}_{2^{w-1}\text{ links}}.$$

Figure 5 illustrates the case $w = 2$, with subsequent generators ( ), (3, 4), (1, 2), (1, 3) (2, 4) and (2, 3), generating subsequent subgroups of orders 1, 2, 4, 8 and 24. Table 2 illustrates the case $w = 3$, giving the subsequent generators both by their permutation notation and by the boolean expressions of their outputs. In the latter, we use the short-hand notation $\overline{X}$ for NOT X. The last column gives the order of the generated subgroup. The reader will easily link this table to figure 1.

Note that a convenient set of $w - 1$ generators for the subchain $\mathbf{C}_w \subset \cdots \subset \mathbf{R}_w$ is provided by a set of strong generators of the subgroup of exchangers. Exchangers are defined as gates with truth tables merely consisting of an exchange of columns. Thus the outputs $P_1, P_2, \ldots, P_w$ are a permutation of the inputs $A_1, A_2, \ldots, A_w$. Figure 2(b) gives an example for $w = 3$, where $P_1 = A_1$, $P_2 = A_3$ and $P_3 = A_2$, i.e. where the second and third inputs are permuted. Note that the permutation notation of this gate is not (2, 3), but (2, 3) (6, 7). In order to distinguish permutations of columns of a truth table from permutations of its rows, we will denote column permutations with a semicolon instead of a comma. Thus we write

(2; 3) = (2, 3) (6, 7).

Table 2. The generators of $\mathbf{R}_3$, using minterms followed by exchangers.

	P_1	P_2	P_3	#
()	A_1	A_2	A_3	$1 = 1$
(7,8)	A_1	A_2	$(A_1$ AND $A_2)$ XOR A_3	$2 = 2$
(5, 6)	A_1	A_2	$(A_1$ AND $\overline{A_2})$ XOR A_3	$2^2 = 4$
(3, 4)	A_1	A_2	$(\overline{A_1}$ AND $A_2)$ XOR A_3	$2^3 = 8$
(1, 2)	A_1	A_2	$(\overline{A_1}$ AND $\overline{A_2})$ XOR A_3	$2^4 = 16$
(5, 7) (6, 8)	A_1	A_1 XOR A_2	A_3	$2^5 = 32$
(1, 3) (2, 4)	A_1	$\overline{A_1}$ XOR A_2	A_3	$2^6 = 64$
(1, 5) (2, 6) (3, 7) (4, 8)	$\overline{A_1}$	A_2	A_3	$2^7 = 128$
(1; 2) = (3, 5) (4, 6)	A_2	A_1	A_3	$2^7 \cdot 3 = 384$
(2; 3) = (2, 3) (6, 7)	A_1	A_3	A_2	$2^7 \cdot 3^2 \cdot 5 \cdot 7 = 40\,320$

Table 3. The generators of $\mathbf{R}_3$, using piterms followed by exchangers.

	P_1	P_2	P_3	#
()	A_1	A_2	A_3	$1 = 1$
(7, 8)	A_1	A_2	$(A_1$ AND $A_2)$ XOR A_3	$2 = 2$
(5, 6) (7, 8)	A_1	A_2	A_1 XOR A_3	$2^2 = 4$
(3, 4) (7, 8)	A_1	A_2	A_2 XOR A_3	$2^3 = 8$
(1, 2) (3, 4) (5, 6) (7, 8)	A_1	A_2	1 XOR A_3	$2^4 = 16$
(5, 7) (6, 8)	A_1	A_1 XOR A_2	A_3	$2^5 = 32$
(1, 3) (2, 4) (5, 7) (6, 8)	A_1	1 XOR A_2	A_3	$2^6 = 64$
(1, 5) (2, 6) (3, 7) (4, 8)	1 XOR A_1	A_2	A_3	$2^7 = 128$
(1; 2) = (3, 5) (4, 6)	A_2	A_1	A_3	$2^7.3 = 384$
(2; 3) = (2, 3) (6, 7)	A_1	A_3	A_2	$2^7 \cdot 3^2 \cdot 5 \cdot 7 = 40\,320$

One should be aware that this equality is only valid within the $w = 3$ framework. Indeed, if $w = 2$, then (2; 3) is meaningless, whereas (2; 3) = (3, 5) (4, 6) (11, 13) (12, 14) for $w = 4$. The permutation notation of an arbitrary exchange gate has the form of a product of 2^{w-2} disjoint transpositions. The exchangers (together with the identity gate) form a group $\mathbf{E}_w$ of order $w!$, which is a subgroup [24–26] of $\mathbf{R}_w$, and is isomorphic to the symmetric group $\mathbf{S}_w$.

We remind that the subgroup $\mathbf{C}_w$ only allows permutations within the duos $\{1, 2\}, \{3, 4\}, \ldots, \{2^w - 1, 2^w\}$. Adding the generator (1; 2) to $\mathbf{C}_w$ allows also some permutations *between* the four quarters $\{1, 2, \ldots, 2^{w-2}\}, \{2^{w-2} + 1, 2^{w-2} + 2, \ldots, 2^{w-1}\}, \{2^{w-1} + 1, 2^{w-1} + 2, \ldots, 2^{w-1} + 2^{w-2}\}$ and $\{2^{w-1} + 2^{w-2} + 1, 2^{w-1} + 2^{w-2} + 2, \ldots, 2^w\}$; the next generator, i.e. (2; 3), introduces permutations *between* the eight eighths; etc, up to generator (w − 1; w). In our example, $\mathbf{S}_2$ wr ($\mathbf{S}_2$ wr $\mathbf{S}_2$) is thus inflated subsequently to $\mathbf{S}_2$ wr $\mathbf{S}_4$ and to $\mathbf{S}_8$.

Table 3 is the variant of table 2, the minterm expansions being replaced by the corresponding Reed–Muller expansions.

It should be stressed that the two chains of maximal subgroups of $\mathbf{R}_w$, we have generated above, are, by far, not the only possible chains of maximal subgroups. For example, the partial chain between $\mathbf{C}_w$ and $\mathbf{R}_w$ can be replaced by another one, resulting from choosing $a = w - 1$, i.e. from dividing the set of elements $\{1, 2, 3, \ldots, 2^w\}$ into two halves: $\{\{1, 2, \ldots, 2^{w-1}\}, \{2^{w-1} + 1, 2^{w-1} + 2, \ldots, 2^w\}\}$, leading to $\mathbf{S}_{2^{w-1}}$ wr $\mathbf{S}_2$, of order $2^{x(w)} \cdot [y(w-1)]^2$. Proceeding further to ($\mathbf{S}_{2^{w-2}}$ wr $\mathbf{S}_2$) wr $\mathbf{S}_2$, etc, this eventually leads to

Table 4. The generators of $\mathbf{R}_3$, using exchangers before minterms.

	P_1	P_2	P_3	#
()	A_1	A_2	A_3	$1 = 1$
(2; 3) = (2, 3) (6, 7)	A_1	A_3	A_2	$2 = 2$
(1; 2) = (3, 5) (4, 6)	A_2	A_1	A_3	$2 \cdot 3 = 6$
(1, 5) (2, 6) (3, 7) (4, 8)	$\overline{A_1}$	A_2	A_3	$2^4 \cdot 3 = 48$
(5, 7) (6, 8)	A_1	A_1 XOR A_2	A_3	$2^6 \cdot 3 \cdot 7 = 1344$
(7, 8)	A_1	A_2	(A_1 AND A_2) XOR A_3	$2^7 \cdot 3^2 \cdot 5 \cdot 7 = 40\,320$

subgroups of subsequent orders $2^{x(w)} \cdot y(w)$, $2^{x(w)} \cdot [y(w-1)]^2$, $2^{x(w)} \cdot [y(w-2)]^4$, ..., $2^{x(w)} \cdot [y(2)]^{2^{w-2}}$ and $2^{x(w)} \cdot [y(1)]^{2^{w-1}} = 2^{x(w)}$.

In order to generate such a chain, we take the same strong generators of $\mathbf{S}_w$, but in the opposite order, i.e. first (w − 1; w), then (w − 2; w − 1), etc, up to (1; 2), introducing subsequently permutation *within* quartettes, *within* octets, etc. In our example $w = 3$, if we first introduce generator (2; 3) and then generator (1; 2), then the order of the last but one subgroup is indeed $2^7 \cdot 3^2 = 1152$, instead of $2^7 \cdot 3 = 384$, as in table 2. Thus here ($\mathbf{S}_2$ wr $\mathbf{S}_2$) wr $\mathbf{S}_2$ is subsequently inflated to $\mathbf{S}_4$ wr $\mathbf{S}_2$ and to $\mathbf{S}_8$.

6. Cheap generating sets

In sections 4 and 5, we generated the group $\mathbf{R}_w$ of reversible logic gates of width w in a mathematically 'natural' way. In technology, however, things go differently. Indeed, for technical implementations, different elements of $\mathbf{R}_w$ do not have the same 'price'. Some gates are easier to implement in hardware than others. Therefore, in practical situations, a designer tries to generate the whole group with as much 'cheap' generators as possible.

In any technology the subgroup $\mathbf{E}_w$ is easy to implement, as it performs only routing of bit streams. In many technologies, also the group of inverters $\mathbf{I}_w$ can be built at low price. So, in many cases, we may assume that both routings and inversions are easily realizable [27]. This, for example, is the case in r-MOS [28–30], a reversible version of dual-line pass-transistor logic [31], as well as in other recovery logics, such as split-level charge recovery logic (SCRL) [32], energy-recovery logic (ERL) [33] and reversible energy recovery logic (RERL) [34].

An inverter is a gate where an output P_i is either equal to its corresponding input A_i or to latter's inverse NOT A_i. The inverters of width w form a subgroup [25] of $\mathbf{R}_w$ of order 2^w, which is isomorphic to $\mathbf{Z}_2^w$. Figure 2(*c*) gives an example for $w = 3$, where $P_1 = A_1$, $P_2 = \mathtt{NOT}\ A_2$ and $P_3 = A_3$, i.e. where the second input is inverted. The inverters can also be interpreted as control gates with control functions that are dependent on zero variables: $P_1 = 0$ XOR A_1, $P_2 = 1$ XOR A_2 and $P_3 = 0$ XOR A_3. Next in complexity come simple control gates controlled by one variable, by two variables, etc. We thus build up the whole group $\mathbf{R}_w$ in the following way:

$$\overbrace{\mathbf{I}_w \subset \cdots \subset \mathbf{E}_w}^{w \text{ links}} \subset \overbrace{\mathbf{I}_w : \mathbf{E}_w \subset \cdots \subset \mathbf{R}_w}^{w \text{ links}}.$$

The total chain consists of $2w$ links. The subgroup $\mathbf{F}_w = \mathbf{I}_w : \mathbf{E}_w$ is the semidirect product of $\mathbf{I}_w$ and $\mathbf{E}_w$ and has been introduced before [25]. It has $w!2^w$ elements.

Table 4 illustrates the case $w = 3$. The reader will note that, after the introduction of the exchange generators, the three Feynman primitives are added. Comparison with table 2 reveals that now the odd prime factors of $(2^w)!$ start appearing before all factors 2 are generated.

Table 5. The generators of $\mathbf{R}_3$, using exchangers before one minterm.

	P_1	P_2	P_3	#
()	A_1	A_2	A_3	$1 = 1$
(2; 3) = (2, 3) (6, 7)	A_1	A_3	A_2	$2^1 = 2$
(1; 2) = (3, 5) (4, 6)	A_2	A_1	A_3	$2^1 \cdot 3 = 6$
(1, 5) (2, 6) (3, 7) (4, 8)	$\overline{A_1}$	A_2	A_3	$2^4 \cdot 3 = 48$
(7, 8)	A_1	A_2	$(A_1$ AND $A_2)$ XOR A_3	$2^7 \cdot 3^2 \cdot 5 \cdot 7 = 40\,320$

Instead of enlarging the subset $\mathbf{E}_w$ with subsequently more expensive CONTROLLED^i NOTs (with i subsequently equal to 0, 1, 2, ..., $w-1$), one can also enlarge it to the full group $\mathbf{R}_w$ in only two steps, by adding to the exchangers only the NOT and the CONTROLLED^{w-1} NOT. The resulting chain of subgroups $\mathbf{1}_w \subset \cdots \subset \mathbf{E}_w \subset \mathbf{F}_w \subset \mathbf{R}_w$ has $w+2$ links. This is illustrated in table 5, for the case $w = 3$. The fact that the exchangers, the NOT, and the CONTROLLED^{w-1} NOT suffice to generate the whole group of reversible gates of width w is proved by Toffoli [8]. An independent proof, by induction on w, can be found in [35].

How to choose between a generating set as given in table 4 and a set of generators as given in table 5 depends on the application. The latter choice has the advantage of a small library of building blocks, the former choice needs a larger library ($w-2$ more basic blocks), but can make some gates (e.g., $192-48 = 144$ gates in our example $w = 3$) in a cheaper way (e.g., with fewer transistors and thus with less expensive silicon area).

We remark that the w strong generators for generating the $w!$ exchangers can be replaced by two generators. Indeed, the symmetric group $\mathbf{S}_w$ can be generated [24, 36] by one transposition (w − 1; w) and the w-fold cycle (1; 2; 3; ...; w − 1; w). This leads to the short subgroup chain $\mathbf{1}_w \subset \mathbf{e}_w \subset \mathbf{E}_w \subset \mathbf{F}_w \subset \mathbf{R}_w$, where $\mathbf{e}_w$ is the subgroup of degree 2 consisting of the w-bit follower and the w-bit gate where the last two bits are exchanged. The chain has subsequent orders 1, 2, $w!$, $w!2^w$ and $(2^w)!$.

One could finally remark that not only $\mathbf{S}_w$, but also $\mathbf{S}_{2^w}$ can be generated by only two generators, leading to the minimum length chain $\mathbf{1}_w \subset \mathbf{f}_w \subset \mathbf{R}_w$, with subsequent orders 1, 2 and $(2^w)!$. However, for practical applications, we lack a simple implementation of the generator $(1, 2, 3, \ldots, 2^w - 1, 2^w)$.

7. Even generating sets

As mentioned incidentally in section 5, the alternating group $\mathbf{A}_{2^w}$ is a maximal subgroup of $\mathbf{S}_{2^w}$. We define the even reversible gates as the subgroup of the reversible gates consisting of all gates that are represented by an even permutation. This subgroup is, of course, isomorphic with the alternating group. None of the subgroup chains described in sections 5 and 6 contains the group of even gates. In the example of table 2, the reason is obvious: the odd generators (7, 8), (5, 6), etc, make it impossible that the last subgroup (before $\mathbf{R}_3$) in the chain would be even. In tables 4 and 5, all generators, except the last one, are even. Nevertheless the last but one subgroup in the subgroup chain is not isomorphic to $\mathbf{A}_8$.

A simple way to construct a chain of subgroups containing the subgroup of even gates as the last but one link is as follows. We first introduce the generators of the exchange group, then one inverter. As in section 6, this yields subgroup $\mathbf{F}_w$ of order $w!2^w$. Then we add the generator $(2^{w-1} - 1, 2^{w-1})\ (2^w - 1, 2^w)$, i.e. the gate that realises

$$P_i = A_i \qquad \text{for all} \quad i \in \{1, 2, \ldots, w-1\}$$
$$P_w = (A_2 \text{ AND } A_3 \text{ AND} \cdots \text{AND } A_{w-1}) \text{ XOR } A_w.$$

For convenience, we call this gate the CONTROLLED^{w-2} NOT gate, however not without stressing that here it is a gate of width w, i.e. not of width $w-1$. This additional generator enlarges $\mathbf{F}_w$ to the subgroup of even gates. Toffoli [8] mentions that $\mathbf{F}_w$ is inflated to a subgroup of the group of even gates. The proof that this subgroup is the subgroup of even gates itself, can be found in the appendix. Next, adding any odd generator suffices to inflate the group of even gates to the group $\mathbf{R}_w$ of all gates. However, it is surprising that this procedure is only valid for widths w of four or larger. Indeed:

- For $w = 2$, the CONTROLLED^{w-2} NOT is nothing else but the NOT, i.e. the inverter. The exchangers, together with the inverter, generate the subgroup $\mathbf{I}_2 : \mathbf{E}_2$ of order $2!2^2 = 8$, whereas $\mathbf{A}_4$ is of order $4!/2 = 12$.
- For $w = 3$, the CONTROLLED^{w-2} NOT is the CONTROLLED NOT. The exchangers, together with the NOT and the CONTROLLED NOT, generate a subgroup of order 1344 (see table 4), whereas $\mathbf{A}_8$ is of order $8!/2 = 20\,160$.

A variant of the last generator set, resulting in exactly the same subgroup chain, is found by replacing the CONTROLLED^{w-2} NOT by the CONTROLLED^2 NOT:

$$P_i = A_i \qquad \text{for all} \quad i \in \{1, 2, \ldots, w-1\}$$
$$P_w = (A_{w-2}\ \texttt{AND}\ A_{w-1})\ \texttt{XOR}\ A_w$$

with permutation notation $(7, 8)\,(15, 16)\,(23, 24)\cdots(2^w - 1, 2^w)$. This fact is proved by Toffoli [8] and independently by Raa [35]. Also this theorem is only valid for $w \geqslant 4$. Indeed,

- for $w = 2$, the CONTROLLED^2 NOT is meaningless;
- for $w = 3$, the exchangers, together with the NOT and the CONTROLLED^2 NOT, generate the full group $\mathbf{R}_3$ of order 40 320 (see table 5), whereas $\mathbf{A}_8$ is of order 20 160.

8. Beyond binary logic

In the above discussions, we have implicitly assumed that digital logic is a synonym for binary logic. However, there also exist logics based on digital numbers that can have r different values. The integer r is called the radix of the logic. Besides the well-known case $r = 2$ (binary logic), also the case $r = 3$ (ternary logic) is applied in digital computing [37–40]. We will restrict ourselves here to prime radices.

It is clear that the results in previous sections can easily be extrapolated towards arbitrary prime radix p. The group of reversible logic gates is now of order $(p^w)!$. Its Sylow p-subgroups of order $p^{x(w)}$, with

$$x(w) = p^{w-1} + p^{w-2} + \cdots + p + 1 = \frac{p^w - 1}{p - 1}$$

will now play the central role in the classification of the subgroups. The role of $\mathbf{Z}_2$ is taken over by $\mathbf{Z}_p$.

For $r = 3$, figure 6 shows two different NOT gates and two different CONTROLLED NOT gates. The two NOTs, together with the one-trit follower, form a group isomorphic with $\mathbf{Z}_3$, a subgroup of the whole group of reversible one-trit gates (which is a group isomorphic with $\mathbf{S}_3$ and thus of order $3! = 6$). The two CONTROLLED NOTs, together with the two-trit follower, also form a group isomorphic with $\mathbf{Z}_3$, a subgroup of the whole group of reversible two-trit gates (which is a group isomorphic with $\mathbf{S}_{3^2}$ and thus of order $(3^2)! = 9! = 362\,880$).

Al-Rabadi and Perkowski [41] give a detailed study of more complex reversible multi-valued gates.

A	P
0	1
1	2
2	0

(a)

A	P
0	2
1	0
2	1

(b)

A_1A_2	P_1P_2
0 0	0 0
0 1	0 1
0 2	0 2
1 0	1 0
1 1	1 1
1 2	1 2
2 0	2 1
2 1	2 2
2 2	2 0

(c)

A_1A_2	P_1P_2
0 0	0 0
0 1	0 1
0 2	0 2
1 0	1 0
1 1	1 1
1 2	1 2
2 0	2 2
2 1	2 0
2 2	2 1

(d)

Figure 6. Ternary truth tables: (*a*) NOT, (*b*) NOT, (*c*) CONTROLLED NOT and (*d*) CONTROLLED NOT.

9. Conclusion

The group $\mathbf{R}_w$ of reversible binary gates of width w has (except for very small w) very many subgroups and a very complex lattice diagram. From this diagram, we have investigated only a few sequences of subgroups.

First, we have constructed chains with as many links as possible, i.e. where any subgroup is a maximal subgroup of the next. This leads to chains of $2^w + w - 1$ subgroups, where the subgroup $\mathbf{C}_w$ of control gates plays a central role. In ascending order, all indices of the chain equal 2, up to $\mathbf{C}_w$. After that, all indices are odd, up to $\mathbf{R}_w$ itself. The control gates are particularly interesting

- from a mathematical point of view, because they form one of the Sylow 2-subgroups of the entire group $\mathbf{R}_w$, and
- from a technological point of view, because there exist straightforward ways to implement them into hardware, e.g., branches of minterms (see figure 3 and table 2) or branches of piterms (see table 3).

Secondly, we have constructed short chains, which are therefore generated by a small set of generators, which therefore can implement an arbitrary element by cascading building blocks from a small library.

Finally, the reader can easily extrapolate the results of binary logic to any higher radix logic.

Appendix

Definition. *A neighbour 3-cycle of the elements* $\{1, 2, \ldots, n\}$ *is any 3-cycle of the form* $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$*, thus any of the* $n-2$ *cycles* $(1, 2, 3), (2, 3, 4), \ldots, (\mathtt{n}-2, \mathtt{n}-1, \mathtt{n})$.

Lemma 1. *The neighbour 3-cycles generate the group of all even permutations, i.e. the group* $\mathbf{A}_n$.

Lemma 2. *The subgroup of exchangers* $\mathbf{E}_w$*, augmented with the* NOT *gate and the* CONTROLLED^{w-2} NOT *gate, generates the group of even simple control gates.*

Theorem. *The subgroup of exchangers* $\mathbf{E}_w$*, augmented with the* NOT *gate and the* CONTROLLED^{w-2} NOT *gate, generates the group of even reversible gates.*

The proof is by induction on w:

1. We prove that the theorem is true for $w = 4$: this fact has been checked with the help of computer, using the computer algebra package GAP [42].
2. We assume that the theorem is true for $w = W$ and prove it to be true for $w = W + 1$. For this purpose, we prove that the exchangers, the NOT, and the CONTROLLED^{W-1} NOT generate the neighbour 3-cycles $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$ for all i satisfying $1 \leqslant i \leqslant 2^{W+1} - 2$. Four cases have to be distinguished:
   - 2.1. The case $1 \leqslant i \leqslant 2^W - 2$. We make a gate of width $W + 1$ by combining a follower of unitary width ($P_1 = A_1$) combined with a gate of width W (with inputs $A_2, A_3, \ldots, A_{w+1}$ and outputs $P_2, P_3, \ldots, P_{w+1}$) and permutation $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$. The latter can be generated because
     - the CONTROLLED^{W-2} NOT can be synthesized because of lemma 2 and
     - the exchangers, the NOT and this CONTROLLED^{W-2} NOT, suffice to generate $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$ because of the induction hypothesis.

     The combined gate does not have a neighbour 3-cycle as its permutation, but is represented by the permutation $p = (\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)\,(\mathtt{i}+2^W, \mathtt{i}+1+2^W, \mathtt{i}+2+2^W)$. We cascade four gates of width $W + 1$, generating the permutation $p^{-1}cpc$, where c is an even simple control gate. The gate p^{-1} can be synthesized because it equals p^2; the gate c can be synthesized because of lemma 2. We choose
     - $c = (\mathtt{i}, \mathtt{i}+1)\,(\mathtt{i}+4, \mathtt{i}+5)$, if i is odd,
     - $c = (\mathtt{i}+1, \mathtt{i}+2)\,(\mathtt{i}+3, \mathtt{i}+4)$, if i is even.

     In both subcases we find that $p^{-1}cpc$ equals $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$.
   - 2.2. The case $2^W + 1 \leqslant i \leqslant 2^{W+1} - 2$. This case is analogous to case 2.1.
   - 2.3. The case $i = 2^W - 1$. Because of case 2.1, we can generate the neighbour 3-cycle $t = (2^{W-2} - 1, 2^{W-2}, 2^{W-2} + 1)$. We let this gate be preceded by the exchanger e and be followed by the exchanger e^{-1}, with $e = (1; 2; 3)$, which equals a product of 2^{W-1} disjoint 3-cycles $(2^{W-2} + 1, 2^{W-1} + 1, 2^W + 1) \cdots (\ldots, \ldots, \ldots)$. We find $e^{-1}te = (2^{W-2} - 1, 2^{W-2}, 2^W + 1)$. Now applying $2^W - 2^{W-2}$ times the identity
   $$(\mathtt{k}, \mathtt{k}+1, \mathtt{k}+2) \cdot (\mathtt{k}, \mathtt{k}+1, \mathtt{l}) \cdot (\mathtt{k}, \mathtt{k}+1, \mathtt{k}+2)^{-1} = (\mathtt{k}+1, \mathtt{k}+2, \mathtt{l})$$
   with $l = 2^W + 1$ and with k subsequently equal to $2^{W-2} - 1, 2^{W-2}, \ldots$ and $2^W - 2$, finally yields the 3-cycle $(2^W - 1, 2^W, 2^W + 1)$, i.e. the neighbour 3-cycle $(\mathtt{i}, \mathtt{i}+1, \mathtt{i}+2)$ searched for.
   - 2.4. The case $i = 2^W$. This case is analogous to case 2.3.
3. By virtue of steps 1 and 2, the subgroup of exchangers $\mathbf{E}_w$, augmented with the NOT gate and the CONTROLLED^{w-2} NOT gate, for any $w \geqslant 4$, generates all the neighbour 3-cycles of the elements $\{1, 2, \ldots, 2^w\}$, and thus, by virtue of lemma 1, all possible even permutations of $\{1, 2, \ldots, 2^w\}$.

References

[1] Bennett C 1973 Logical reversibility of computation *IBM J. Res. Dev.* **17** 525–32
[2] Bennett C and Landauer R 1985 The fundamental physical limits of computation *Sci. Am.* **253** (July) 38–46

[3] Keyes R and Landauer R 1970 Minimal energy dissipation in logic *IBM J. Res. Dev.* **14** 153–7
[4] Landauer R 1961 Irreversibility and heat generation in the computational process *IBM J. Res. Dev.* **5** 183–91
[5] Stix G 1998 Riding the back of electrons *Sci. Am.* **279** (September) 20–1
[6] Feynman R 1985 Quantum mechanical computers *Opt. News* **11** 11–20
[7] Feynman R 1996 *Feynman Lectures on Computation* ed A Hey and R Allen (Reading, MA: Addison-Wesley)
[8] Toffoli T 1981 Bicontinuous extensions of invertible combinatorial functions *Math. Syst. Theory* **14** 13–23
[9] Barenco A, Bennett C, Cleve R, Di Vincenzo D, Margolus N, Shor P, Sleator T, Smolin J and Weinfurter H 1995 Elementary gates for quantum computation *Phys. Rev.* A **52** 3457–67
[10] Beckman D, Chari A, Devabhaktuni S and Preskill J 1996 Efficient networks for quantum factoring *Phys. Rev.* A **54** 1034–63
[11] De Vos A, Desoete B, Janiak F and Nogawski A 2001 Control gates for reversible computers *Proc. 11th Int. Workshop on Power and Timing Modeling, Optimization and Simulation (Yverdon, Sept. 2001)* 9.2.1–9.2.10
[12] Desoete B and De Vos A A reversible carry-look-ahead adder using control gates *Integr. VLSI J.* in press
[13] Nève A and Flandre D 2001 Branch-based logic for high performance carry-select adders in 0.25 μm bulk silicon-on-insulator CMOS technologies *Proc. 11th Int. Workshop on Power and Timing Modeling, Optimization and Simulation (Yverdon, Sept. 2001)* 8.2.1–8.2.10
[14] Cárdenas H and Lluis E 1965 El normalizador del p-grupo de Sylow del grupo simétrico S_{p^n} *Bol. Soc. Mat. Mex.* **9** 1–6
[15] Kaloujnine L 1948 La structure des p-groupes de Sylow des groupes symétriques finis *Ann. Sci. École Normale Supérieure* **65** 239–76
[16] Kaloujnine L 1951 Ob odnom obobshenie Silovskich p-podgrup simmetrizjkisch grup *Acta Math. Acad. Sci. Hung.* **2** 197–221
[17] Weir A 1955 The Sylow subgroups of the symmetric groups *Proc. Am. Math. Soc.* **6** 534–41
[18] Teschke L 1979 Über die Normalteiler der p-Sylowgruppe der symmetrischen Gruppe von Grade p^m *Math. Nachr.* **87** 197–212
[19] Hall M 1991 *The Theory of Groups* (Providence, RI: AMS Chelsea) pp 82–3
[20] Butler G 1991 *Fundamental Algorithms for Permutation Groups* (Berlin: Springer) pp 78–87
[21] Green D 1986 *Modern Logic Design* (Workingham: Addison-Wesley) pp 133–64
[22] Aschbacher M and Scott L 1985 Maximal subgroups of finite groups *J. Algebr.* **92** 44–80
[23] Liebeck M, Praeger C and Saxl J 1987 A classification of the maximal subgroups of the finite alternating and symmetric groups *J. Algebr.* **111** 365–83
[24] Rayner M and Newman D 1995 On the symmetry of logic *J. Phys. A: Math. Gen.* **28** 5623–31
[25] Storme L, De Vos A and Jacobs G 1999 Group theoretical aspects of reversible logic gates *J. Universal Comput. Sci.* **5** 307–21
[26] Kerntopf P 2000 On the efficiency of reversible (3,3)-gates *Proc. 7th Int. Conf. on Mixed Design of Integrated Circuits and Systems (Gdynia, June 2000)* pp 185–90
[27] Perkowski M *et al* 2001 A general decomposition for reversible logic *Proc. 2001 Reed–Muller Workshop (Starkville, Aug. 2001)* pp 119–38
[28] De Vos A 1996 Introduction to r-MOS systems *Proc. 4th Workshop on Physics and Computation (Boston, Nov. 1996)* pp 92–6
[29] De Vos A 1997 Towards reversible digital computers *Proc. European Conf. on Circuit Theory and Design (Budapest, Sept. 1997)* pp 923–31
[30] De Vos A 1999 Reversible computing *Prog. Quantum Electron.* **23** 1–49
[31] Zimmermann R and Fichtner W 1997 Low-power logic styles: CMOS versus pass-transistor logic *IEEE J. Solid-State Circuits* **32** 1079–90
[32] Younis S and Knight T 1994 Asymptotically zero energy split-level charge recovery logic *Proc. IEEE Int. Workshop on Low Power Design (Napa Valley, April 1994)* pp 177–82
[33] Athas W, Tzartzanis N, Svensson L and Peterson L 1997 A low-power microprocessor based on resonant energy *IEEE J. Solid-State Circuits* **32** 1693–701
[34] Lim J, Kim D and Chae S 1999 A 16-bit carry-lookahead adder using reversible energy recovery logic for ultra-low-energy systems *IEEE J. Solid-State Circuits* **34** 898–903
[35] Raa B 2001 Reversibele logische poorten met behulp van controlepoorten *MSc Thesis* Universiteit Gent, Gent
[36] Stewart I 1989 *Galois Theory* (London: Chapman and Hall) p 120
[37] Yoeli M and Rosenfeld G 1965 Logical design of ternary switching circuits *IEEE Trans. Electron. Comput.* **14** 19–29
[38] Hurst S 1984 Multiple-valued logic—its status and its future *IEEE Trans. Comput.* **33** 1160–79

[39] Butler J 1991 *Multiple-valued Logic in VLSI* (New York: IEEE)
[40] Mateo D and Rubio A 1998 Design and implementation of a 5 × 5 trits multiplier in a quasi-adiabatic ternary CMOS logic *IEEE J. Solid-State Circuits* **33** 1111–6
[41] Al-Rabadi A and Perkowski M 2001 New classes of multi-valued reversible decompositions for three-dimensional layout *Proc. 2001 Reed–Muller Workshop (Starkville, Aug. 2001)* pp 185–204
[42] Schönert M 1992 `GAP` *Comput. Algebr. Nede. Nieuwsbrief* **9** 19–28