The decomposition of an arbitrary reversible logic circuit

J. Phys. A: Math. Gen. **39** (2006) 5015–5035 doi:10.1088/0305-4470/39/18/017

The decomposition of an arbitrary reversible logic circuit

Alexis De Vos, Yvan Van Rentergem and Koen De Keyser

Imec v z w and Vakgroep elektronika en informatiesystemen, Universiteit Gent,
Sint Pietersnieuwstraat 41, B-9000 Gent, Belgium

Received 9 September 2005
Published 19 April 2006
Online at stacks.iop.org/JPhysA/39/5015

Abstract

The $(2^w)!$ reversible logic circuits of width w, i.e. reversible logic circuits with w inputs and w outputs, together with the action of cascading, form a group $\mathbf{G}$, isomorphic to the symmetric group $\mathbf{S}_{2^w}$. We define two conjugate subgroups $\mathbf{G}_1$ and $\mathbf{G}_2$. Together they partition the group $\mathbf{G}$ into $2^{w-1}+1$ double cosets. These allow us to decompose an arbitrary member of $\mathbf{G}$ into a cascade of three simpler members. This decomposition is a far relative of the well-known LU decomposition of a square matrix.

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

1. Introduction

Reversible computing [1, 2] is useful both in lossless classical computing [3–5] and in quantum computing [6]. For its study, we consider all reversible logic circuits of width w, i.e. with w binary inputs $A_1, A_2, \ldots, A_w$ and w binary outputs $P_1, P_2, \ldots, P_w$; see figure 1. Indeed, for reversible logic circuits, the number of outputs necessarily equals the number of inputs. The truth table of such a logic circuit corresponds to a permutation of the 2^w input rows $(0, 0, \ldots, 0, 0)$, $(0, 0, \ldots, 0, 1)$, $\ldots$, $(1, 1, \ldots, 1, 1)$ of the table. An example (with $w = 3$) is given in table 1(a). Given a particular reversible circuit (e.g. by explicit supply of such truth table), the question arises how to implement it into hardware.

Recently, several synthesis methods have been presented in the literature. Van Rentergem *et al* [7, 8] have presented a synthesis method based on group theory. Indeed, all $(2^w)!$ permutations of 2^w objects form a group with respect to the operation of composition. Therefore, reversible circuits form a group with respect to the operation of cascading. The group of all reversible logic circuits with equal width w form a group, which we will denote by $\mathbf{G}$. It is isomorphic to the symmetric group $\mathbf{S}_{2^w}$. Van Rentergem *et al* consider a subset of $\mathbf{G}$: they consider all reversible circuits which satisfy $P_1 = A_1$. Those form a subgroup $\mathbf{G}_1$, isomorphic to the direct product group $\mathbf{S}_{2^{w-1}} \times \mathbf{S}_{2^{w-1}}$ of order $(2^{w-1}!)^2$. This subgroup is a special case of a Young subgroup. Indeed, any subgroup of the form $\mathbf{S}_{n_1} \times \mathbf{S}_{n_2} \times \cdots \times \mathbf{S}_{n_k}$

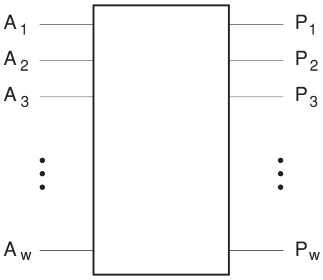

Figure 1. Arbitrary reversible logic circuit.

Table 1. Truth table of some reversible circuits: (a) arbitrary circuit, (b) linear circuit, (c) homogeneous linear circuit.

(a)		(b)		(c)	
$A_1A_2A_3$	$P_1P_2P_3$	$A_1A_2A_3$	$P_1P_2P_3$	$A_1A_2A_3$	$P_1P_2P_3$
0 0 0	1 1 1	0 0 0	1 0 0	0 0 0	0 0 0
0 0 1	1 0 1	0 0 1	0 0 0	0 0 1	1 0 0
0 1 0	1 0 0	0 1 0	0 0 1	0 1 0	1 0 1
0 1 1	0 0 0	0 1 1	1 0 1	0 1 1	0 0 1
1 0 0	1 1 0	1 0 0	1 1 1	1 0 0	0 1 1
1 0 1	0 1 0	1 0 1	0 1 1	1 0 1	1 1 1
1 1 0	0 0 1	1 1 0	0 1 0	1 1 0	1 1 0
1 1 1	0 1 1	1 1 1	1 1 0	1 1 1	0 1 1

(where $\{n_1, n_2, \ldots, n_k\}$ forms a partition of n, i.e. where $n_1 + n_2 + \cdots + n_k = n$) is called a Young subgroup [9] of $\mathbf{S}_n$.

If a is a particular circuit of $\mathbf{G}$ and $\mathbf{H}$ is some subgroup of $\mathbf{G}$, then the set of products b_1ab_2 is called the double coset of a. Here, we allow both b_1 and b_2 to equal subsequently all members of $\mathbf{H}$. The double coset of a is denoted by $\mathbf{H}a\mathbf{H}$. The double cosets partition the supergroup $\mathbf{G}$. Indeed, each element of $\mathbf{G}$ belongs to one and only one double coset. The different double cosets of $\mathbf{G}$ are not of equal size. The maximal size is h^2, where h is the order of the subgroup $\mathbf{H}$. The minimal size is h. In order to obtain a 'cheap' synthesis, as many circuits as possible should belong to the double coset $\mathbf{H}i\mathbf{H}$ of the identity gate i. This gate (also known as the follower) satisfies the rule

$$\begin{aligned} P_1 &= A_1 \\ P_2 &= A_2 \\ P_3 &= A_3 \\ \cdots &\quad \cdots \\ P_w &= A_w. \end{aligned}$$

Unfortunately, the double coset of the identity gate is of size h, i.e. of minimum size,

The number of double cosets in which $\mathbf{G}$ is partitioned by its subgroup $\mathbf{G}_1$ is $2^{w-1} + 1$. We label them with the numbers $0, 1, 2, \ldots, 2^{w-1}$. How can we find out to which of these double cosets a particular circuit a of $\mathbf{G}$ belongs? It suffices to verify in the truth table how many different inputs $0, A_2, A_3, \ldots, A_w$ give rise to an output $1, P_2, P_3, \ldots, P_w$ (and thus also the number of different inputs $1, A_2, A_3, \ldots, A_w$ which give rise to an output

Table 2. Expanded truth table.

$A_1A_2A_3$	$F_1F_2F_3$	$J_1J_2J_3$	$P_1P_2P_3$
0 0 0	**0** 0 1	**1** 0 1	1 1 1
0 0 1	**0** 1 0	**1** 1 0	1 0 1
0 1 0	**0** 1 1	**1** 1 1	1 0 0
0 1 1	**0** *0 0*	**0** *0 0*	0 0 0
1 0 0	**1** *0 0*	**1** *0 0*	1 1 0
1 0 1	**1** 0 1	**0** 0 1	0 1 0
1 1 0	**1** 1 0	**0** 1 0	0 0 1
1 1 1	**1** 1 1	**0** 1 1	0 1 1

$0, P_2, P_3, \ldots, P_w)$. We can also say that the double coset number equals half the weight of the function $A_1 \oplus P_1(A_1, A_2, A_3, \ldots, A_w)$. Here, the weight of a boolean function $f(A_1, A_2, \ldots, A_w)$ is defined as the number of 1s in its truth table; thus it is an integer in the range 0 to 2^w. Note that, according to this numbering, the double coset $\mathbf{G}_1 i \mathbf{G}_1$ has label 0. If $\mathbf{G}$ is the group of reversible circuits of width 3, then its $8! = 40\,320$ elements are spread over $2^2 + 1 = 5$ double cosets. The circuit of table 1(a) belongs to double coset number 3. We finally note that $\frac{1}{2}$ weight$(A_1 \oplus P_1)$ can be interpreted as a distance between the two columns A_1 and P_1 of the truth table.

We illustrate the Van Rentergem–De Vos–Storme synthesis method for the circuit of table 1(a). We add two extra columns to table 1(a), resulting in table 2. The extra columns F and J are filled in, in five steps:

- first, we fill in $F_1 = A_1$ and $J_1 = P_1$;
- then, on the rows where $F_1 = J_1 = 0$, we give $(F_2, F_3, \ldots, F_w) = (J_2, J_3, \ldots, J_w)$ the first values in the lexicographic ordering, i.e. $(0, 0, \ldots, 0)$ etc;
- then, on the rows where $F_1 = J_1 = 1$, we give $(F_2, F_3, \ldots, F_w) = (J_2, J_3, \ldots, J_w)$ the same first values in the lexicographic ordering;
- then, on the rows where $F_1 = 0$ and $J_1 = 1$, we give $(F_2, F_3, \ldots, F_w) = (J_2, J_3, \ldots, J_w)$ the remaining values in the lexicographic ordering, ending with $(1, 1, \ldots, 1)$;
- finally, on the rows where $F_1 = 1$ and $J_1 = 0$, we give $(F_2, F_3, \ldots, F_w) = (J_2, J_3, \ldots, J_w)$ the same last values in the lexicographic ordering.

Note that the first step is displayed in bold face in table 2, whereas the second and third steps are emphasized in italic.

The above procedure yields a decomposition of the logic circuit into three logic circuits, given in table 3. We see that, automatically, table 3(b) is an (upside-down) simple control gate (as it fixes all but the first column). Tables 3(a) and (c) both are members of $\mathbf{G}_1$ (as they fix the first column); see figure 2(*a*).

Simple control gates are reversible circuits which satisfy the following relationship between outputs and inputs:

$$\begin{aligned}
P_1 &= A_1 \\
P_2 &= A_2 \\
P_3 &= A_3 \\
\cdots &= \cdots \\
P_{w-1} &= A_{w-1} \\
P_w &= f(A_1, A_2, \ldots, A_{w-1}) \oplus A_w.
\end{aligned}$$

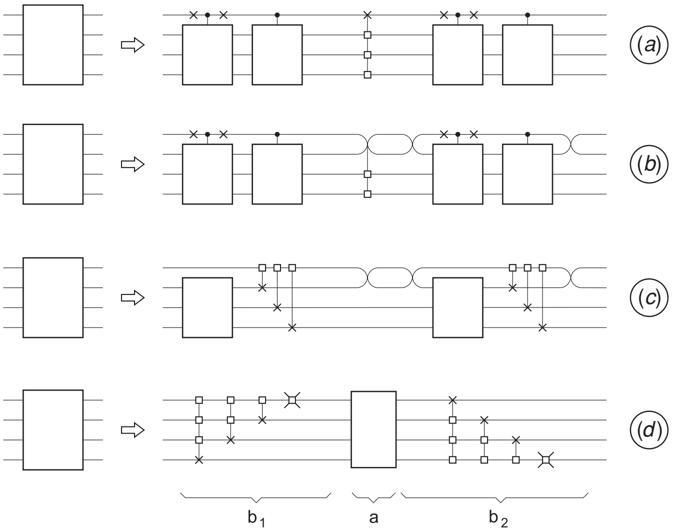

Figure 2. Decomposition of a reversible circuit: (*a*) an arbitrary reversible circuit with the help of one subgroup $\mathbf{G}_1$, (*b*) an arbitrary reversible circuit with the help of two conjugate subgroups $\mathbf{G}_1$ and $\mathbf{G}_2$, (*c*) a homogeneous linear reversible circuit with the help of two conjugate subgroups, (*d*) an arbitrary reversible circuit with the help of two conjugate subgroups **C** and **D**.

Table 3. Decomposed truth table.

(a)		(b)		(c)	
$A_1A_2A_3$	$F_1F_2F_3$	$F_1F_2F_3$	$J_1J_2J_3$	$J_1J_2J_3$	$P_1P_2P_3$
0 0 0	0 0 1	0 0 0	0 0 0	0 0 0	0 0 0
0 0 1	0 1 0	0 0 1	1 0 1	0 0 1	0 1 0
0 1 0	0 1 1	0 1 0	1 1 0	0 1 0	0 0 1
0 1 1	0 0 0	0 1 1	1 1 1	0 1 1	0 1 1
1 0 0	1 0 0	1 0 0	1 0 0	1 0 0	1 1 0
1 0 1	1 0 1	1 0 1	0 0 1	1 0 1	1 1 1
1 1 0	1 1 0	1 1 0	0 1 0	1 1 0	1 0 1
1 1 1	1 1 1	1 1 1	0 1 1	1 1 1	1 0 0

Here f is an arbitrary boolean function of $w-1$ boolean variables. The simple control gates form a group, isomorphic to $\mathbf{S}_2^{2^{w-1}}$, with order $2^{2^{w-1}}$. An upside-down simple control gate obeys

$$\begin{aligned}
P_1 &= f(A_2, A_3, \ldots, A_w) \oplus A_1\\
P_2 &= A_2\\
P_3 &= A_3\\
\cdots &= \cdots\\
P_{w-1} &= A_{w-1}\\
P_w &= A_w.
\end{aligned}$$

Figure 3(*a*) displays the symbol of this controlled inverter. If, in particular, the control function f is a compact Maitra term [7], then we call such gate a Maitra-controlled `NOT` gate or

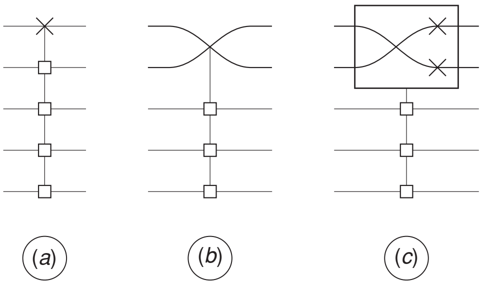

Figure 3. Symbols for upside-down controlled gates: (*a*) controlled NOT, (*b*) controlled SWAP and (*c*) controlled INVERTED SWAP.

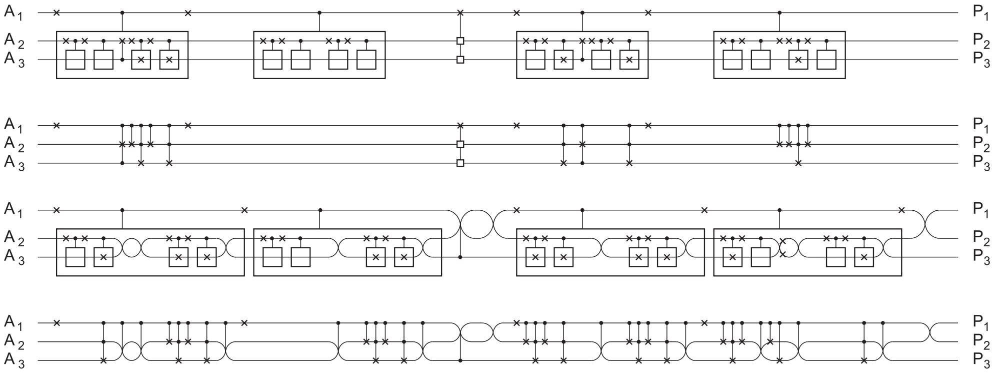

Figure 4. Synthesis of reversible circuit: (*a*) and (*b*) with one subgroup, (*c*) and (*d*) with two conjugate subgroups.

Maitra-controlled inverter. In our example (table 3(b)), the inversion of bit F_1 is controlled by the Maitra term $f(F_2, F_3) = F_2 + F_3$.

Now we apply the above procedure to four small circuits of width $w - 1$: to the upper half of table 3(a), to the lower half of table 3(a), to the upper half of table 3(c), and to the lower half of table 3(c). And so on: we have to apply the procedure to ever more circuits of ever smaller width, until all blocks are of unitary width. Each recursive step gives rise to one or more simple control gates. The result is shown in figure 4(*a*). Such a diagram can immediately be implemented into hardware [8]. Alternatively, it can be translated into a more conventional schematic; see figure 4(*b*). The latter can subsequently be simplified, e.g., with the template technique [10].

Each simple control gate is a representative of a double coset. If we are lucky, some of these representatives are equal to the identity gate i. Hardware implementation of this gate is trivial and gratis. It is just a bus of w wires. However, the probability, that gate i will show up many times during the synthesis procedure, is small, because its double coset is small. Indeed, the double coset $\mathbf{H}i\mathbf{H}$ of i is the smallest of all. It has the minimal size h:

$$y = h$$

Here, y denotes the size of $\mathbf{H}i\mathbf{H}$. We note that the double coset $\mathbf{H}i\mathbf{H}$ is a group: it equals $\mathbf{H}$. Indeed, if b_1 and b_2 both are elements of $\mathbf{H}$, then $b_1ib_2 = b_1b_2$ is also a member of $\mathbf{H}$; conversely if b is an element of $\mathbf{H}$, then there exists a product b_1ib_2 (with both b_1 and b_2 belonging to $\mathbf{H}$) equal to b, because it suffices to choose e.g. $b_1 = b$ and $b_2 = i$. All the other double cosets are not a group. Suffice it to note that they contain no identity element.

It is a pity that the double coset of i is the smallest among all double cosets. Indeed, it means that only a small subset of elements of $\mathbf{G}$ can be written as b_1b_2 (i.e. a decomposition into two factors). All other elements a of $\mathbf{G}$ have to be written as b_1rb_2 (i.e. a decomposition into three factors), where r is a representative of the double coset $\mathbf{H}a\mathbf{H}$. Therefore, in the following section, we present a new synthesis method, where the double coset of the identity is much larger. This should lead automatically to more compact decompositions and thus to cheaper synthesis.

2. The synthesis method

We now define two subgroups of $\mathbf{G}$. We first consider all reversible logic circuits which satisfy $P_1 = A_1$. We again denote this subgroup by $\mathbf{G}_1$. We subsequently consider all reversible logic circuits which satisfy $P_2 = A_2$. We denote this subgroup by $\mathbf{G}_2$. Both subgroups are isomorphic to $\mathbf{S}_{2^{w-1}} \times \mathbf{S}_{2^{w-1}}$. The two subgroups are conjugate. This means that any member b_2 of $\mathbf{G}_2$ can be written as eb_1e^{-1}, where b_1 is an appropriate member of $\mathbf{G}_1$ and where e is a particular member of $\mathbf{G}$. The reader will easily verify that here $e = e^{-1}$ is a special exchanger e_{12}, i.e. the gate:

$$\begin{aligned} P_1 &= A_2 \\ P_2 &= A_1 \\ P_3 &= A_3 \\ \cdots & \quad \cdots \\ P_w &= A_w. \end{aligned}$$

Now, we again take advantage of the powerful tool of double cosets, but now based on two different subgroups. Let $\mathbf{H}_1$ and $\mathbf{H}_2$ be two subgroups of a group $\mathbf{G}$ and a is a particular element of $\mathbf{G}$; then the set of products b_1ab_2 is called the double coset of a and is denoted by $\mathbf{H}_1a\mathbf{H}_2$. Here, we allow b_1 to equal subsequently all members of $\mathbf{H}_1$ and b_2 to equal subsequently all members of $\mathbf{H}_2$. The double cosets partition the supergroup $\mathbf{G}$. Indeed, each element of $\mathbf{G}$ belongs to one and only one double coset. The different double cosets of $\mathbf{G}$ are not of equal size. In general, we can only set a lower and an upper bound to the size. If there exists an element a of $\mathbf{G}$, such that all products b_1ab_2 are different, then the size of its double coset is maximal and equal to h_1h_2, where h_1 is the order of $\mathbf{H}_1$ and h_2 is the order of $\mathbf{H}_2$. Appendix A calculates the size of the double coset $\mathbf{H}_1i\mathbf{H}_2$ of the identity gate i. Let h_{12} be the order of the subgroup formed by the intersection $\mathbf{H}_{12}$ of $\mathbf{H}_1$ and $\mathbf{H}_2$; see figure 5. If y denotes the size of the double coset $\mathbf{H}_1i\mathbf{H}_2$ of i, then we have

$$y = \frac{h_1h_2}{h_{12}}.$$

The smaller is h_{12}, the larger is the double coset $\mathbf{H}_1i\mathbf{H}_2$. The synthesis method, described in [7, 8] and illustrated in section 1, is however based on two identical subgroups $\mathbf{H}_1$ and $\mathbf{H}_2$ (i.e. $\mathbf{H}_1 = \mathbf{H}_2 = \mathbf{G}_1$), such that $h_1 = h_2 = h_{12}$ (say h), and y is only equal to h:

$$y = [(2^{w-1})!]^2.$$

In the present section, we choose two different subgroups $\mathbf{H}_1$ and $\mathbf{H}_2$, i.e. $\mathbf{H}_1 = \mathbf{G}_1$ and $\mathbf{H}_2 = \mathbf{G}_2$. We still have $h_1 = h_2 = (2^{w-1}!)^2$, but now the size of the overlap $\mathbf{H}_{12}$ is small. The subgroup $\mathbf{H}_{12}$ consists of all reversible circuits simultaneously satisfying $P_1 = A_1$ and $P_2 = A_2$. It is isomorphic to $\mathbf{S}_{2^{w-2}} \times \mathbf{S}_{2^{w-2}} \times \mathbf{S}_{2^{w-2}} \times \mathbf{S}_{2^{w-2}}$ and has order $h_{12} = (2^{w-2}!)^4$. Table 4 gives the values of the order g of the group $\mathbf{G}$, the order h of the two subgroups, the order h_{12} of the overlap, and the resulting size y of the double coset $\mathbf{G}_1i\mathbf{G}_2$ of i, as a function

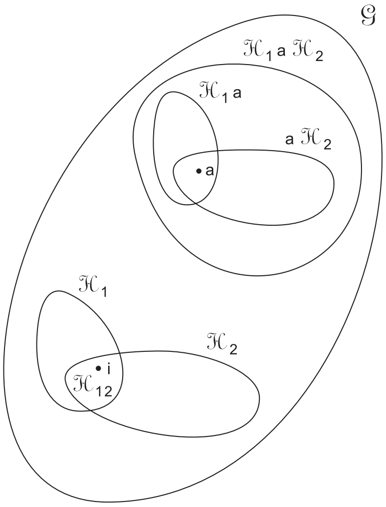

Figure 5. A group and two of its subgroups.

Table 4. The reversible logic circuits: the order g of the whole group, the order $h_1 = h_2 = h$ of the two subgroups, the order h_{12} of the intersection of the two subgroups, and the size y of the double coset of the identity element.

w	$g = 2^w!$	$h = (2^{w-1}!)^2$	$h_{12} = (2^{w-2}!)^4$	$y = [(2^{w-1})!/(2^{w-2})!]^4$
2	24	4	1	16
3	40 320	576	16	20 736
4	20 922 789 888 000	1625 702 400	331 776	7965 941 760 000

of the width w of the circuits. We see that y is indeed much larger than h, such that the new method will be more powerful:

$$y = \left[\frac{(2^{w-1})!}{(2^{w-2})!}\right]^4 .$$

The fact that y is no divisor of g illustrates that (in contrast to $\mathbf{G}_1 i \mathbf{G}_1$ and $\mathbf{G}_2 i \mathbf{G}_2$) $\mathbf{G}_1 i \mathbf{G}_2$ is not a subgroup of $\mathbf{G}$.

The number of double cosets in which $\mathbf{G}$ is partitioned by its subgroups $\mathbf{G}_1$ and $\mathbf{G}_2$ is $2^{w-1} + 1$, i.e. the same number as in section 1. This is no surprise; see appendix B. We again label these equivalence classes with the numbers $0, 1, 2, \ldots, 2^{w-1}$. How can we now find out to which of these double cosets a particular circuit a of $\mathbf{G}$ belongs? By extrapolation of appendix C, we find the following 'classifying functional':

$$\tfrac{1}{2}\ \text{weight}(A_1 \oplus P_2).$$

Note that, according to this numbering, the double coset $\mathbf{G}_1 i \mathbf{G}_2$ has label 2^{w-2}. The circuit of table 1(a) belongs to double coset number 1.

We illustrate the new synthesis method for the example circuit (table 1(a)). We add two extra columns to table 1(a), resulting in table 5. The extra columns F and J are filled in, in six steps:

- first, we fill in $F_1 = A_1$ and $J_2 = P_2$;
- then, we fill in the upper half-column F_2 with the balance of the upper half-column J_2;

Table 5. Expanded truth table.

$A_1A_2A_3$	$F_1F_2F_3$	$J_1J_2J_3$	$P_1P_2P_3$
0 0 0	$\mathbf{0}\ \underline{1}\ 0$	$\underline{0}\ \mathbf{1}\ 0$	1 1 1
0 0 1	$\mathbf{0}\ \underline{0}\ \mathit{0}$	$\underline{0}\ \mathbf{0}\ \mathit{0}$	1 0 1
0 1 0	$\mathbf{0}\ \underline{0}\ \mathit{1}$	$\underline{0}\ \mathbf{0}\ \mathit{1}$	1 0 0
0 1 1	$\mathbf{0}\ \underline{1}\ 1$	$\underline{1}\ \mathbf{0}\ 1$	0 0 0
1 0 0	$\mathbf{1}\ \underline{1}\ 0$	$\underline{1}\ \mathbf{1}\ 0$	1 1 0
1 0 1	$\mathbf{1}\ \underline{1}\ 1$	$\underline{1}\ \mathbf{1}\ 1$	0 1 0
1 1 0	$\mathbf{1}\ \underline{0}\ 0$	$\underline{1}\ \mathbf{0}\ 0$	0 0 1
1 1 1	$\mathbf{1}\ \underline{0}\ 1$	$\underline{0}\ \mathbf{1}\ 1$	0 1 1

- then, we fill in the lower half-column F_2 with the balance of the lower half-column J_2;
- then, we fill in column J_1 with the value of $F_1 \oplus F_2 \oplus J_2$;
- then, we fill in the remaining upper-half columns:
  - first, on the rows where $F_1 \oplus F_2$ equals the majority of the upper half of J_2, we give $(F_3, F_4, \ldots, F_w) = (J_3, J_4, \ldots, J_w)$ the values in the lexicographic ordering, i.e. $(0, 0, \ldots, 0), \ldots, (1, 1, \ldots, 1)$;
  - then, on the rows where $F_1 \oplus F_2$ equals the minority of the upper half of J_2:
    - ∗ where F_2 equals J_2, we give $(F_3, F_4, \ldots, F_w) = (J_3, J_4, \ldots, J_w)$ the first values in the lexicographic ordering, i.e. $(0, 0, \ldots, 0)$ etc;
    - ∗ where F_2 equals $\overline{J_2}$, we give $(F_3, F_4, \ldots, F_w) = (J_3, J_4, \ldots, J_w)$ the last values in the lexicographic ordering, ending with $(1, 1, \ldots, 1)$;
- then, we fill in the remaining lower-half columns, analogously as its upper-halves.

Here, 'majority', 'minority' and 'balance' have the following meaning. Assume an arbitrary string of $2n$ binary numbers $(N_1, N_2, \ldots, N_{2n})$. If the string contains more zeros than ones, then its majority equals 0, else it equals 1. The minority is the inverse of the majority. Now we construct a new string $(M_1, M_2, \ldots, M_{2n})$ by choosing

- $M_1 = N_1, M_2 = N_2, \ldots, M_i = N_i$, where i is the smallest number such that $(M_1, M_2, \ldots, M_i)$ contains either exactly n zeros or exactly n ones;
- $M_{i+1} = M_{i+2} = \cdots = M_{2n}$ equal to the minority of $(N_1, N_2, \ldots, N_{2n})$.

The result is a balanced string $(M_1, M_2, \ldots, M_{2n})$, i.e. a string with n zeros and n ones.

Note that the first step of the procedure is displayed in bold face in table 5, whereas the second, third and fourth steps are underlined and the first substep of the fifth step is emphasized in italic.

The above procedure yields a decomposition of the logic circuit into three logic circuits, given in table 6; see also figure 2(*b*). We see that, automatically, table 6(b) is a controlled gate with two controlled wires (as it fixes all but the first and the second columns). The fourth step in the algorithm guarantees that

$$F_1 \oplus F_2 \oplus J_1 \oplus J_2 = 0.$$

The end note of appendix C then guarantees that the controlled gate is

- either a follower;
- or a controlled exchange of two wires;
- or a controlled exchange-plus-inversion of two wires;
- or a controlled inversion of two wires.

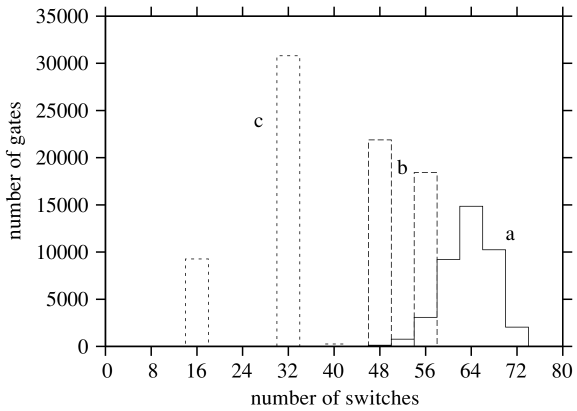

Figure 6. Statistics of switch cost when synthesizing all reversible circuits of $w = 3$: (*a*) with the method of section 1, (*b*) with the method of section 2 and (*c*) with the 'practical' method of appendix E.

Table 6. Decomposed truth table.

(a)		(b)		(c)	
$A_1A_2A_3$	$F_1F_2F_3$	$F_1F_2F_3$	$J_1J_2J_3$	$J_1J_2J_3$	$P_1P_2P_3$
0 0 0	0 1 0	0 0 0	0 0 0	0 0 0	1 0 1
0 0 1	0 0 0	0 0 1	0 0 1	0 0 1	1 0 0
0 1 0	0 0 1	0 1 0	0 1 0	0 1 0	1 1 1
0 1 1	0 1 1	0 1 1	1 0 1	0 1 1	0 1 1
1 0 0	1 1 0	1 0 0	1 0 0	1 0 0	0 0 1
1 0 1	1 1 1	1 0 1	0 1 1	1 0 1	0 0 0
1 1 0	1 0 0	1 1 0	1 1 0	1 1 0	1 1 0
1 1 1	1 0 1	1 1 1	1 1 1	1 1 1	0 1 0

However, the last possibility is excluded by steps 2 and 3 of the algorithm. Figures 3(*b*) and (*c*) show the symbol for the controlled exchange and the controlled exchange-plus-inversion, respectively. In our example, we obtain a Maitra-controlled `SWAP` gate, i.e. a Maitra-controlled exchanger, bits F_1 and F_2 being exchanged iff $F_3 = 1$. The control function thus is the Maitra term $f(F_3) = F_3$. Table 6(a) is a member of $\mathbf{G}_1$ (as the synthesis algorithm's first step guarantees that it fixes the first column); table 6(c) is a member of $\mathbf{G}_2$ (as the algorithm's first step guarantees that it fixes the second column).

Recursive application of the procedure yields the synthesis method. For the example circuit of table 1(a) this leads to the result shown in figure 4(*c*). Translated to a more conventional schematic, this gives figure 4(*d*). The latter can again easily be simplified [11].

In table 4, we see that indeed $y \gg h$. Even more: y is of the order of magnitude of g. It turns out that now the double coset of the identity is the largest (instead of the smallest) of all double cosets in which the group **G** is partitioned. The fact that the double coset of i has a maximum size is not completely a surprise; see appendix D. The property promises attractive possibilities for synthesis.

Figure 6(*b*) shows a statistical distribution for all 40 320 reversible circuits of width w equal 3. The abscissa is the number s of switches needed in the electronic implementation. This number varies from 48 to 56 with an average of 51.7. For comparison, figure 6(*a*) shows similar results for the synthesis method of section 1: switch costs from 48 to 72, with an average of 63.6. For $w = 4$, the synthesis method of the present section (i.e. the two-conjugate-subgroups method) yields a distribution between $s = 216$ and $s = 264$ with an average at $s \approx 240$, whereas the method of section 1 (i.e. the two-equal-subgroups method) yields a distribution ranging between $s = 216$ and $s = 324$ with an average at $s \approx 290$. We stress here that these numerical results are obtained without application of any post-synthesis circuit simplifications, such as template matching or fusion of equal subcircuits.

Together, the two synthesis methods, the former from section 1 and the latter from the present section, lay at the basis of an even more powerful method, described in appendix E.

3. Homogeneous linear reversible circuits

An important subgroup of the group of reversible logic circuits is formed by the linear reversible circuits [12, 13]. A reversible circuit is linear if-and-only-if each of its outputs $P_1, P_2, \ldots, P_w$ is a linear function of the inputs $A_1, A_2, \ldots, A_w$. A function $P(A_1, A_2, \ldots, A_w)$ is linear if its Reed–Muller expansion contains no terms (called piterms) with two-or-more literals. For example, table 1(a) is not linear because it expresses the following functions:

$$P_1 = 1 \oplus A_1A_2 \oplus A_1A_3 \oplus A_2A_3$$
$$P_2 = 1 \oplus A_1 \oplus A_2 \oplus A_1A_2 \oplus A_1A_3$$
$$P_3 = 1 \oplus A_2 \oplus A_3 \oplus A_1A_2 \oplus A_2A_3.$$

It suffices to remark that $P_1(A_1, A_2, A_3)$ is nonlinear, e.g. because of the presence of the term A_1A_2. In contrast, table 1(b) is linear, as all outputs are linear functions of the inputs:

$$P_1 = 1 \oplus A_2 \oplus A_3 \qquad P_2 = A_1 \qquad P_3 = A_1 \oplus A_2.$$

A subgroup of the group of linear reversible circuits is the group of homogeneous linear reversible circuits. A linear reversible circuit is called homogeneous if-and-only-if each of its outputs is a homogeneous function of the inputs. A linear function $P(A_1, A_2, \ldots, A_w)$ is called homogeneous if-and-only-if all the piterms of its Reed–Muller expansion contain exactly one literal. For example, table 1(b) is not homogeneous because of term 1 in the expansion of P_1. In contrast, table 1(c) is homogeneous:

$$P_1 = A_2 \oplus A_3 \qquad P_2 = A_1 \qquad P_3 = A_1 \oplus A_2.$$

The expression can be written in matrix notation:

$$\begin{pmatrix} P_1 \\ P_2 \\ \cdots \\ P_w \end{pmatrix} = \mathbf{M} \begin{pmatrix} A_1 \\ A_2 \\ \cdots \\ A_w \end{pmatrix}.$$

The square matrix $\mathbf{M}$ has a dimension $w \times w$ and has a determinant equal to 1. In our example, we have

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}.$$

The group of homogeneous linear reversible circuits is isomorphic with the general linear group GL$(w, 2)$ of order $2^{(w-1)w/2} w!_2$, where $w!_2$ is the bifactorial of w, the q-factorial being a generalization of the ordinary factorial $w! = w!_1$:

$$w!_q = 1(1 + q)(1 + q + q^2) \cdots (1 + q + \cdots + q^{w-1}).$$

We now define two subgroups:

Table 7. The homogeneous linear reversible logic circuits: the order g of the whole group, the order $h_1 = h_2 = h$ of the two subgroups, the order h_{12} of the intersection of the two subgroups, and the size y of the double coset of the identity element.

	g	h	h_{12}	$y = 2^{(w^2-w+2)/2}.$
w	$= 2^{(w-1)w/2}w!_2$	$= 2^{(w-1)w/2}(w-1)!_2$	$= 2^{(w-2)(w+1)/2}(w-2)!_2$	$(2^{w-2}-1).(w-2)!_2$
2	6	2	1	4
3	168	24	4	144
4	20 160	1344	96	18 816
5	9999 360	322 560	10 752	9676 800
6	20 158 709 760	319 979 520	5160 960	19 838 730 240

Table 8. The homogeneous linear reversible logic circuits: the order g of the whole group, the order $h_1 = h_2 = h$ of the two subgroups, the order h_{12} of the intersection of the two subgroups, and the size y of the double coset of the identity element.

w	$g = 2^{(w-1)w/2}w!_2$	$h = 2^{(w-1)w/2}$	$h_{12} = 1$	$y = 2^{(w-1)w}$
2	6	2	1	4
3	168	8	1	64
4	20 160	64	1	4096
5	9999 360	1024	1	1048 576
6	20 158 709 760	32 768	1	1073 741 824

- all homogeneous linear reversible circuits satisfying $P_1 = A_1$ and
- all homogeneous linear reversible circuits satisfying $P_2 = A_2$.

Both are isomorphic to the indirect product $\mathrm{GL}(w-1, 2)$**:**$\mathbf{S}_2^{w-1}$ of order $2^{(w-1)w/2}(w-1)!_2$.

In table 7, we again see how $y \gg h$. Now, we have only two double cosets: a smaller one with size h and a larger one with size $y = g - h$. Thus, again $\mathbf{H}_1 i \mathbf{H}_2$ is 'the largest of all double cosets'. This means that 'a lot' of homogeneous linear reversible circuits are simply a cascade of one fixing the first bit and another one fixing the second bit. Only 'a few' need a third circuit in the middle: a `SWAP` gate; see figure 2(c).

4. The LU decomposition

The decomposition presented in the previous section, is similar to the well-known LU matrix decomposition: $\mathbf{M} = \mathbf{LU}$ with $\mathbf{L}$ a lower triangular matrix and $\mathbf{U}$ an upper triangular matrix. This is no surprise: the $(w \times w)$ matrices of type $\mathbf{L}$ form a subgroup of the $(w \times w)$ matrices $\mathbf{M}$. Its order is $2^{(w-1)w/2}$. The $\mathbf{U}$ matrices form a second subgroup, conjugate to the one of the $\mathbf{L}$ matrices. Indeed, any $\mathbf{U}$ can be written as the product $\mathbf{ELE}^{\,-1}$, where $\mathbf{L}$ is the transpose of $\mathbf{U}$ and where $\mathbf{E} = \mathbf{E}^{-1}$ is the 'miror matrix', i.e. the matrix with all elements equals 0, except for those on the second diagonal. In fact, the $\mathbf{L}$ and the $\mathbf{U}$ subgroups are two of the $w!_2$ Sylow 2-subgroups of the general linear group. The intersection of the subgroups $\mathbf{L}$ and $\mathbf{U}$ consists of the trivial subgroup with a single element, i.e. the identity matrix $\mathbf{I}$. Because $h_{12} = 1$, we have $y = h_1 h_2 = h^2$.

Table 8 shows that the double coset $\mathbf{L}i\mathbf{U}$ of matrix $\mathbf{I}$ is large: $y = h^2 \gg h$. That is exactly the reason why a large number of matrices can in fact be LU decomposed. After all, it is because $h \ll g$, that people neither apply LL decomposition nor UU decomposition. Not only the double coset $\mathbf{L}i\mathbf{U}$ is large, it also is the largest among all $w!$ double cosets.

The matrices **M** which do not belong to the double coset of the identity matrix cannot be LU decomposed. They can be written as **LRU**, where **R** is a representative of the double coset to which **M** belongs. These matrices **R** play a similar role as the pivot matrices in the theory of LU decomposition. The example matrix in section 3 is one that cannot be LU decomposed. The following decompositions exists

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

but

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

works equally well.

Now the question arises what is the equivalent of the LU decomposition of homogeneous linear reversible circuits in the group of arbitrary reversible circuits? The generalization of the group **L** is formed by the group **C** of the so-called control gates [14, 15]. Control gates are reversible circuits which satisfy the following relationship between outputs and inputs:

$$\begin{aligned}P_1 &= f_1(.)\oplus A_1\\ P_2 &= f_2(A_1)\oplus A_2\\ P_3 &= f_3(A_1,A_2)\oplus A_3\\ \cdots &= \cdots\\ P_w &= f_w(A_1,A_2,\ldots,A_{w-1})\oplus A_w.\end{aligned}$$

Here f_i is an arbitrary boolean function of $i-1$ boolean variables. The group is isomorphic to $\mathbf{S}_2^{2^w-1}$, has order 2^{2^w-1}, and is a Sylow 2-subgroup of the group of all reversible circuits of width w. The generalization of the group **U** is the conjugate group $e\mathbf{C}e^{-1}$, which we will denote by **D**. Here $e=e^{-1}$ denotes a special exchanger: the mirror gate

$$\begin{aligned}P_1 &= A_w\\ P_2 &= A_{w-1}\\ P_3 &= A_{w-2}\\ \cdots &\quad \cdots\\ P_w &= A_1.\end{aligned}$$

Note how bit line w here plays a role similar to the role played by the second bitline in section 2. Thus the elements of **D** obey

$$\begin{aligned}P_1 &= f_1(A_2,A_3,\ldots,A_w)\oplus A_1\\ P_2 &= f_2(A_3,\ldots,A_w)\oplus A_2\\ \cdots &= \cdots\\ P_{w-1} &= f_{w-1}(A_w)\oplus A_{w-1}\\ P_w &= f_w(.)\oplus A_w.\end{aligned}$$

Figure 2(d) shows the decomposition of an arbitrary circuit of **G** into the cascade b_1ab_2, with b_1 a member of **C** and b_2 a member of **D**.

The intersection $\mathbf{H}_{12}$ of the two subgroups is the subgroup of inverters, isomorphic to $\mathbf{S}_2^w$, of order 2^w. Table 9 shows that once again the double coset size y is much larger than the

Table 9. The reversible logic circuits: the order g of the whole group, the order $h_1 = h_2 = h$ of the two subgroups, the order h_{12} of the intersection of the two subgroups, and the size y of the double coset of the identity element.

w	$g = 2^w!$	$h = 2^{2^w-1}$	$h_{12} = 2^w$	$y = 2^{2^{w+1}-w-2}$
2	24	8	4	16
3	40 320	128	8	2048
4	20 922 789 888 000	32 768	16	67 108 864

subgroup order h. Unfortunately, the size y of the double coset of the identity gate is small compared to g, the order of the whole group. That is why **C** and **D** do not form a good choice for partitioning the supergroup **G**. Comparison of the y values of table 9 with the y values of table 4 leads to the conclusion that the groups $\mathbf{G}_1$ and $\mathbf{G}_2$ in section 2 are much more powerful. Sections 3 and 4 are presented only for illustrating the relationship between our decomposition of reversible circuits (section 2) and the LU decomposition of matrices.

5. Conclusion

We have presented a method of synthesizing an arbitrary reversible logic circuit, based on double cosets. In order to make the synthesis 'cheap', we have taken care that the double coset of the identity gate is large. For that purpose, we choose two different subgroups instead of two identical ones. More precisely, we choose two conjugate subgroups. As a result, the hardware cost of the implementation (both for average circuits and worst-case circuits) is reduced by about a factor of 2. It is remarkable that the procedure is a far nephew of the well-known LU decomposition of square matrices. The synthesis procedure leads to the introduction of a powerful tool: the distance matrix **D**. Its elements D_{ij} express how strongly the jth binary output differs from the ith binary input of the reversible circuit.

Acknowledgment

The authors thank Professor Leo Storme (Universiteit Gent) for valuable discussions.

Appendix A. The size of a double coset

In order to calculate the size of the double coset $\mathbf{H}_1 a \mathbf{H}_2$ of an arbitrary circuit a of the group **G**, we have to consider an arbitrary cascade $b_1 a b_2$, where b_1 is a member of the subgroup $\mathbf{H}_1$ and b_2 is a member of the subgroup $\mathbf{H}_2$. If $\mathbf{H}_{12}$ is the intersection of $\mathbf{H}_1$ and $\mathbf{H}_2$, then also $\mathbf{H}_{12}$ is a group; see figure 5. We denote by h_1, h_2 and h_{12} the orders of $\mathbf{H}_1$, $\mathbf{H}_2$ and $\mathbf{H}_{12}$, respectively. Further we denote by j_{12} the size of the intersection $\mathbf{H}_1 a \cap a\mathbf{H}_2$ of the left coset $\mathbf{H}_1 a$ and the right coset $a\mathbf{H}_2$.

As there are h_1 choices for b_1 and h_2 choices for b_2, we have $h_1 h_2$ products $b_1 a b_2$. However, we have to be aware of 'double counting'. Another product, say $b_1' a b_2'$, might have the same value as $b_1 a b_2$. If indeed

$$b_1' a b_2' = b_1 a b_2,$$

then (after multiplying to the left-hand side by b_1^{-1} and to the right-hand side with $b_2'^{-1}$) we necessarily have

$$b_1^{-1} b_1' a = a b_2 b_2'^{-1}.$$

As the lhs is an element of the left coset $\mathbf{H}_1 a$ and the rhs is an element of the right coset $a\mathbf{H}_2$, both sides represent a same element of $\mathbf{H}_1 a \cap a\mathbf{H}_2$, say a_{12}. Therefore, we have:

$$b_1' = b_1 a_{12} a^{-1} \qquad b_2' = a_{12}^{-1} a b_2.$$

Conversely, if we take an arbitrary element a_{12} from $\mathbf{H}_1 a \cap a\mathbf{H}_2$, then,

- because $a_{12} \in \mathbf{H}_1 a$, automatically we have $a_{12}a^{-1} \in \mathbf{H}_1$ and thus $b_1' \in \mathbf{H}_1$;
- analogously, automatically we have $b_2' \in \mathbf{H}_2$; and
- $b_1' a b_2' = b_1 a b_2$.

As there are j_{12} choices for the element a_{12}, there are j_{12} products $b_1' a b_2'$ with the same outcome (i.e. outcome $b_1 a b_2$). Thus among the $h_1 h_2$ products $b_1 a b_2$ there are only

$$y = \frac{h_1 h_2}{j_{12}}$$

products with a distinct value.

As h_1 is not only the order of subgroup $\mathbf{H}_1$ but also the size of the left coset $\mathbf{H}_1 a$, as h_2 is not only the order of subgroup $\mathbf{H}_2$ but also the size of the right coset $a\mathbf{H}_2$, we have $1 \leqslant j_{12} \leqslant \min(h_1, h_2)$, and thus $\max(h_1, h_2) \leqslant y \leqslant h_1 h_2$.

We close this appendix with a few notes:

Note A1. In the special case where $\mathbf{H}_1$ and $\mathbf{H}_2$ are conjugate, we have $h_1 = h_2$ (say h) and therefore

$$y = \frac{h^2}{j_{12}}$$

and $h \leqslant y \leqslant h^2$, where the case $y = h$ is valid iff $\mathbf{H}_1 = \mathbf{H}_2 = \mathbf{H}_{12}$ (i.e. for two equal subgroups), and the case $y = h^2$ is valid iff $\mathbf{H}_{12}$ is the trivial group consisting of the identity gate i alone.

Note A2. In the special case where a equals the identity gate i, we obtain

$$y = \frac{h_1 h_2}{h_{12}}.$$

Note A3. In the above, the number j_{12} is the size of the intersection of two cosets. However, it can also be interpreted as the order of a group. Indeed, the intersection $\mathbf{H}_1 a \cap a\mathbf{H}_2$ has the same size as the intersection $\mathbf{H}_1 \cap a\mathbf{H}_2 a^{-1}$, as well as the same size as the intersection $a^{-1}\mathbf{H}_1 a \cap \mathbf{H}_2$. Because $\mathbf{H}_1$, $\mathbf{H}_2$, $a^{-1}\mathbf{H}_1 a$ and $a\mathbf{H}_2 a^{-1}$ all are subgroups of $\mathbf{G}$, also $\mathbf{H}_1 \cap a\mathbf{H}_2 a^{-1}$ and $a^{-1}\mathbf{H}_1 a \cap \mathbf{H}_2$ are subgroups of $\mathbf{G}$.

Appendix B. Conjugate subgroups

Let $\mathbf{G}$ be an arbitrary group and $\mathbf{H}_1$ an arbitrary subgroup. Let $\mathbf{H}_2$ be a subgroup of $\mathbf{G}$, conjugate to $\mathbf{H}_1$:

$$\mathbf{H}_2 = e\mathbf{H}_1 e^{-1},$$

for some $e \in \mathbf{G}$. Let a be an arbitrary member of $\mathbf{G}$ and b an arbitrary element of its double coset $\mathbf{H}_1 a \mathbf{H}_1$. Thus we have

$$b = hah',$$

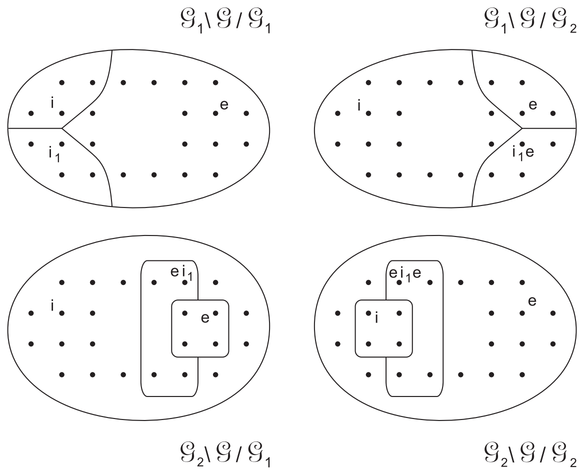

Figure 7. Double coset spaces generated by the group **G** and its two conjugate subgroups $\mathbf{G}_1$ and $\mathbf{G}_2$. Note: i is the identity gate ($P_1 = A_1$, $P_2 = A_2$), e is the exchanger gate ($P_1 = A_2$, $P_2 = A_1$), and i_1 is the inverter ($P_1 = \overline{A_1}$, $P_2 = A_2$).

where both h and h' are appropriate members of $\mathbf{H}_1$. From this equality follow three new equalities:

$$\begin{aligned} be^{-1} &= h(ae^{-1})(eh'e^{-1}) \\ eb &= (ehe^{-1})(ea)h' \\ ebe^{-1} &= (ehe^{-1})(eae^{-1})(eh'e^{-1}). \end{aligned}$$

This shows that

- be^{-1} belongs to the double coset $\mathbf{H}_1(ae^{-1})\mathbf{H}_2$ of ae^{-1},
- eb belongs to the double coset $\mathbf{H}_2(ea)\mathbf{H}_1$ of ea, and
- ebe^{-1} belongs to the double coset $\mathbf{H}_2(eae^{-1})\mathbf{H}_2$ of eae^{-1}.

Conversely, we can demonstrate that

- if c belongs to $\mathbf{H}_1(ae^{-1})\mathbf{H}_2$, then ce belongs to $\mathbf{H}_1 a\mathbf{H}_1$;
- if c belongs to $\mathbf{H}_2(ea)\mathbf{H}_1$, then $e^{-1}c$ belongs to $\mathbf{H}_1 a\mathbf{H}_1$; and
- if c belongs to $\mathbf{H}_2(eae^{-1})\mathbf{H}_2$, then $e^{-1}ce$ belongs to $\mathbf{H}_1 a\mathbf{H}_1$.

This demonstrates that the four double cosets $\mathbf{H}_1 a\mathbf{H}_1, \mathbf{H}_1(ae^{-1})\mathbf{H}_2, \mathbf{H}_2(ea)\mathbf{H}_1$ and $\mathbf{H}_2(eae^{-1})\mathbf{H}_2$ have an equal size.

Therefore, the four double coset spaces $\mathbf{G}_1\backslash\mathbf{G}/\mathbf{G}_1$, $\mathbf{G}_1\backslash\mathbf{G}/\mathbf{G}_2$, $\mathbf{G}_2\backslash\mathbf{G}/\mathbf{G}_1$ and $\mathbf{G}_2\backslash\mathbf{G}/\mathbf{G}_2$ consist of an equal number of distinct double cosets, with an equal size distribution. Figure 7 shows the example where **G** is the group of all reversible circuits of width 2 (isomorphic to $\mathbf{S}_4$ of order 24). The subgroup $\mathbf{H}_1$ is the subgroup $\mathbf{G}_1$ of all circuits that satisfy $P_1 = A_1$ (isomorphic to $\mathbf{S}_2^2$ of order 4). The gate e is the exchanger ($P_1 = A_2$, $P_2 = A_1$), such that $\mathbf{H}_2$ is the subgroup $\mathbf{G}_2$ of circuits satisfying $P_2 = A_2$. We see that all four double coset spaces consist of three double cosets, two of size 4 and one of size 16.

Appendix C. The case w = 2

Let us consider all $4! = 24$ reversible circuits with two inputs (A_1 and A_2) and two outputs (P_1 and P_2). They fall apart into three classes, which we label 0, 1 and 2, respectively.

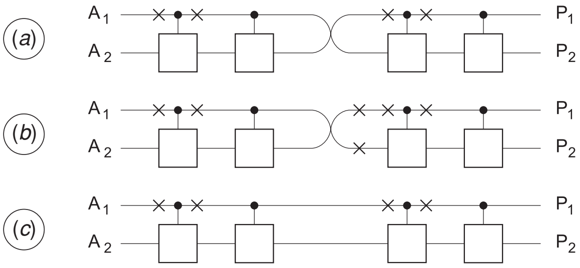

Figure 8. Arbitrary reversible logic circuit of width 2. (*a*) class 0, (*b*) class 2 and (*c*) class 1.

Table 10. Truth table of reversible circuits of width 2 and (*a*) class 0 or (*b*) class 2.

A_1A_2	P_1P_2	A_1A_2	P_1P_2	A_1A_2	P_1P_2	A_1A_2	P_1P_2
			(a)				
0 0	0 0	0 0	1 0	0 0	0 0	0 0	1 0
0 1	1 0	0 1	0 0	0 1	1 0	0 1	0 0
1 0	0 1	1 0	0 1	1 0	1 1	1 0	1 1
1 1	1 1	1 1	1 1	1 1	0 1	1 1	0 1
			(b)				
0 0	1 1	0 0	0 1	0 0	1 1	0 0	0 1
0 1	0 1	0 1	1 1	0 1	0 1	0 1	1 1
1 0	1 0	1 0	1 0	1 0	0 0	1 0	0 0
1 1	0 0	1 1	0 0	1 1	1 0	1 1	1 0

- Class 0 consists of the four truth tables of table 10(a). They form the double coset of the exchanger: figure 8(*a*).
- Class 2 consists of the four truth tables of table 10(b). They form the double coset of the 'inverted exchanger': figure 8(*b*).
- Class 1 consists of the remaining 16 reversible truth tables. They form the double coset of the follower: figure 8(*c*).

In order to distinguish the three classes, we note the following 'classifying functional':

$$W = \tfrac{1}{2}[\text{weight(upper half column } P_2) + \text{weight(lower half column } \overline{P_2})]$$

equaling 0 for class 0, 1 for class 1, and 2 for class 2. Equally well, W can be written as

$$\begin{aligned} W &= \tfrac{1}{2}[\text{weight}(\overline{A_1}P_2) + \text{weight}(A_1\overline{P_2})] \\ &= \tfrac{1}{2}\,\text{weight}(A_1 \oplus P_2), \end{aligned}$$

where now whole columns are taken into account.

Incidentally, we note a useful property. If we calculate the function

$$A_1 \oplus A_2 \oplus P_1(A_1, A_2) \oplus P_2(A_1, A_2),$$

then we find the zero-function only in four of the twenty-four cases:

- a gate from class 1: the follower ($P_1 = A_1$, $P_2 = A_2$),
- one gate of class 0: the exchanger ($P_1 = A_2$, $P_2 = A_1$),
- one gate of class 2: the 'inverted exchanger' ($P_1 = \overline{A_2}$, $P_2 = \overline{A_1}$), and
- a second gate from class 1: the inverter ($P_1 = \overline{A_1}$, $P_2 = \overline{A_2}$).

Appendix D. The largest double coset

We consider the set S of all reversible circuits of width w, where output P_1 is a linear boolean function of the w inputs $A_1, A_2, \ldots, A_w$:

$$P_1 = \epsilon_0 \oplus \epsilon_1 A_1 \oplus \epsilon_2 A_2 \oplus \cdots \oplus \epsilon_w A_w,$$

where $\epsilon_i \in \{0, 1\}$ for all $0 \leqslant i \leqslant w$. There are $2^{w+1}[(2^{w-1})!]^2$ such logic circuits. However, they do not form a group.

By simply calculating the distance $\frac{1}{2}$ weight$(A_1 \oplus P_1)$, we find that the elements of S occupy only three of the $2^{w-1} + 1$ double cosets out of the double coset space $\mathbf{G}_1\backslash\mathbf{G}/\mathbf{G}_1$:

- S contains all $[(2^{w-1})!]^2$ members of the 0th double coset, i.e. those reversible circuits that obey $P_1 = A_1$.
- S contains all $[(2^{w-1})!]^2$ members of the double coset with label 2^{w-1}, i.e. those reversible circuits that obey $P_1 = 1 \oplus A_1 = \overline{A_1}$.
- All other elements of S (i.e. a large majority of S) are members of the double coset numbered 2^{w-2}, i.e. the largest double coset.

In particular, the exchanger e_{12}, which exchanges first and second bits, is an element of S, because of $P_1 = A_2$. It belongs to the maximum-size double coset.

The two subgroups $\mathbf{G}_1$ and $\mathbf{G}_2$ are conjugate: $\mathbf{G}_2 = e_{12}\mathbf{G}_1 e_{12}^{-1}$. Therefore, according to appendix B, the size of $\mathbf{G}_1 a \mathbf{G}_1$ equals the size of $\mathbf{G}_1\left(a e_{12}^{-1}\right)\mathbf{G}_2$. Choosing $a = e_{12}$, we conclude that the size of $\mathbf{G}_1 i \mathbf{G}_2$ equals the size of $\mathbf{G}_1 e_{12} \mathbf{G}_1$. Above, we have seen that the latter equals the largest coset of $\mathbf{G}_1\backslash\mathbf{G}/\mathbf{G}_1$. Thus the double coset $\mathbf{G}_1 i \mathbf{G}_2$ is the largest of all double cosets of $\mathbf{G}_1\backslash\mathbf{G}/\mathbf{G}_2$.

Appendix E. Practical synthesis method

If we have to synthesize a particular reversible circuit a, with the given truth table of width w, we first investigate whether or not it belongs to some $\mathbf{G}_n i \mathbf{G}_n$, where $\mathbf{G}_n$ is the subgroup satisfying $P_n = A_n$. In that case we have the structure of figure 9(a). This is very interesting, as it can be replaced by figure 9(b). Indeed $\mathbf{G}_n i \mathbf{G}_n$ equals $\mathbf{G}_n$. Such circuits we call here type 1. Their number we denote by N_1.

If the circuit belongs to none of the w double cosets $\mathbf{G}_n i \mathbf{G}_n$, we investigate whether or not it belongs to some $\mathbf{G}_n e_{12} \mathbf{G}_m$, where e_{12} is the exchanger of first and second bits. In that case we have the structure of figure 9(c). This is equally interesting, as it can be replaced by figure 9(d). Such circuits we call here type 2. They obey a $P_m = A_n$ rule (with either $m = n$ or $m \neq n$). Their number we denote by N_2. By M_2 we denote the number of circuits which are of type 2 without being of type 1.

If a circuit is neither of type 1 nor of type 2, it is still possible that there is some index n, such that $P_n = \overline{A_n}$. In other words: that it belongs to the double coset $\mathbf{G}_n i_1 \mathbf{G}_n$ of i_1, the inverter of the first bit ($P_1 = \overline{A_1}$, $P_2 = A_2$, $P_3 = A_3, \ldots, P_w = A_w$). In that case we have the structure of figure 9(e). Also this is advantageous, as it can be replaced by figure 9(f). Such circuits we call type 3. We denote their number by N_3. By M_3 we denote the number of circuits of type 3, which are neither of type 1, nor of type 2.

If a circuit does not belong to types 1–3, we investigate whether it is of type $\mathbf{G}_n e_{12} i_1 i_2 \mathbf{G}_m$. In that case we have the structure of figure 9(g). It can be replaced by figure 9(h). It obeys a $P_m = \overline{A_n}$ rule. Gates of this kind we call type 4. There are N_4 circuits of this type; there are M_4 circuits which belong to type 4, without belonging to types 1–3.

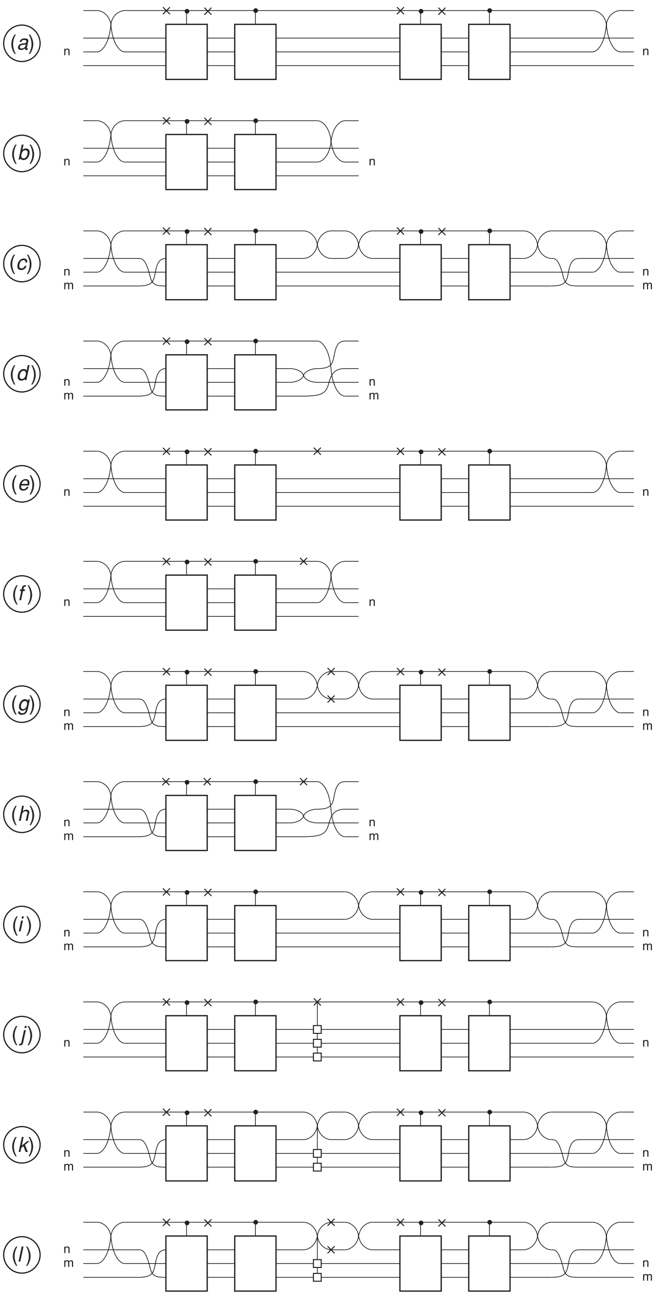

Figure 9. Arbitrary reversible logic circuit: (*a*) type 1, (*b*) type 1 simplified, (*c*) type 2, (*d*) type 2 simplified, (*e*) type 3, (*f*) type 3 simplified, (*g*) type 4 (*h*) type 4 simplified, (*i*) type 5, (*j*) type 6, (*k*) type 7 and (*l*) type 8.

If a circuit belongs to none of types 1–4, it is worth while to look whether it is of type 5: figure 9(*i*). It is then a member of the large double coset $\mathbf{G}_n i \mathbf{G}_m$. We denote by N_5 the number of circuits of type 5 and by M_5 the number of these which belong to none of types 1–4.

Table 11. Number of circuits of width w and type i: (a) total number N_i; (b) number M_i being of type i without being of any type j with $j < i$.

(a)

w	N_1	N_2	N_3	N_4	N_5
2	7	7	7	7	20
3	1681	4902	1681	4902	39 680

(b)

w	M_1	M_2	M_3	M_4	M_5	M_6
2	7	7	5	5	0	0
3	1681	3221	1498	2864	30 416	640

Only the remaining $M_6 = (2^w)! - M_1 - M_2 - M_3 - M_4 - M_5$ circuits need a full structure of type $\mathbf{G}_n r \mathbf{G}_m$ (figures 9(j)–(l)). Here, r is the representative of its double coset. Table 11 gives the numbers N_i and M_i, respectively.

In practice, we construct the distance matrix $\mathbf{D}$:

$$D_{nm} = \tfrac{1}{2}\,\text{weight}(A_n \oplus P_m).$$

For our 3 × 3 example circuit (table 1(a)), we obtain

$$\mathbf{D} = \begin{pmatrix} 3 & 1 & 2 \\ 3 & 3 & 2 \\ 3 & 2 & 2 \end{pmatrix}.$$

If the diagonal of this matrix contains at least one zero, the circuit is of type 1. If the matrix contains at least one off-diagonal zero, then the circuit is of type 2. If the diagonal of the matrix contains at least one element of value 2^{w-1}, the circuit is of type 3. If the off-diagonal elements contain at least one entry equal to 2^{w-1}, the circuit is of type 4. If at least one off-diagonal element has value 2^{w-2}, the circuit is of type 5.

If the distance matrix is of none of these five simple types, then the synthesis contains (in the middle) either a controlled inverter (figure 9(j)), or a controlled exchanger (figure 9(k)), or a controlled 'inverted exchanger' (figure 9(l)). In order to make such a controlled gate as cheap as possible, we proceed as follows: we look subsequently for D_{nm} numbers equal to 2^{w-2}, 2^{w-3}, $2^{w-2} + 2^{w-3}$, 2^{w-4}, etc. In our example, we have $\mathbf{D}_{13} = 2$, such that the circuit is of type 5 with, in the middle, an identity gate. In order to facilitate the search for control functions with as few letters as possible, we calculate a new matrix $\mathbf{M}$ from the matrix elements D_{nm}:

$$M_{nm} = w - 1 - \text{tail}(D_{nm} \text{ modulo } 2^{w-1}).$$

Here, the 'tail' of an integer is the number of zeros at the right end of its binary notation. Note that tail(0) $= 2^{w-2}$, whereas, for $x > 0$, the function tail(x) equals the exponent of 2 in the prime factorization of x. In our example, we obtain

$$\mathbf{M} = \begin{pmatrix} 2 & 2 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix}.$$

If a circuit is of type 6, we look for off-diagonal elements M_{nm} as small as possible. In our example, this is e.g. $\mathbf{M}_{13} = 1$, yielding a Maitra term with only one letter.

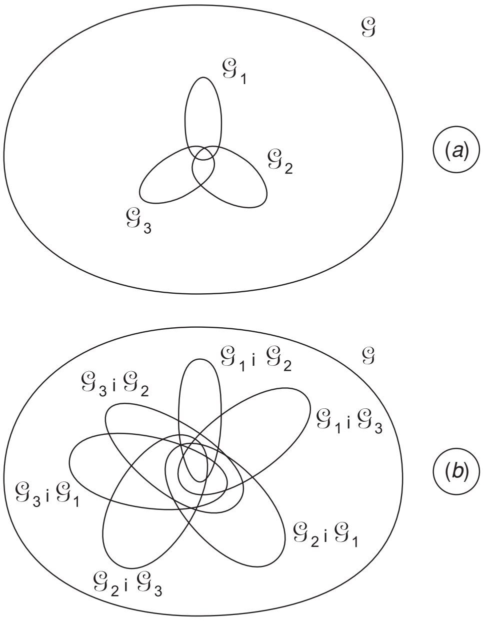

Figure 10. The group of reversible circuits of width 3 and the set of circuits belonging to (*a*) type 1 and (*b*) type 5.

When the first iteration (with width w) is finished, we start all over again, by applying the same procedure (for width $(w-1)$) to either two (figures 9(*b*), (*d*), (*f*) or (*h*)) or four (figures 9(*i*)–(*l*)) subcircuits, and so on, until all subcircuits are of unitary width.

Figure 6(*c*) gives the switch cost in the case we apply the 'practical' algorithm to all reversible circuits with $w = 3$. We see that the 'practical' synthesis method leads to a distribution ranging from $s = 16$ to only $s = 40$ with an average as small as $s \approx 28.4$. For $w = 4$, the present method yields a cost distribution between $s = 44$ and $s = 196$, with a peak at $s = 152$.

Figure 10(*a*) shows the w subgroups $\mathbf{G}_n$ in the case $w = 3$. These three conjugate subgroups form what is called in set theory a flower. If a circuit belongs to this flower, it is of type 1. Because of the generalized inclusion–exclusion principle, the number of circuits in this flower is

$$N_1 = \sum_{i=1}^{w} (-1)^{i-1} C_w^i (2^{w-i}!)^{2^i}.$$

Analogously, we have

$$N_2 = \sum_{i=1}^{w} (-1)^{i-1} i! \left(C_w^i\right)^2 (2^{w-i}!)^{2^i}.$$

The $w(w-1)$ double cosets $\mathbf{G}_m i \mathbf{G}_n$ also form a flower: figure 10(*b*). Not only does this flower have more petals, it also has larger petals. This explains why N_5 is so much larger than N_1.

References

[1] Markov I 2003 An introduction to reversible circuits *Proc. 12th Int. Workshop on Logic and Synthesis (Laguna Beach, May 2003)* pp 318–9

[2] Frank M 2005 Introduction to reversible computing: motivation, progress and challenges *Proc. 2005 Computing Frontiers Conference (Ischia, May 2005)* pp 385–90

[3] Wayner P 1994 Silicon in reverse *Byte* **19** 67–74

[4] De Vos A 2003 Lossless computing *Proc. IEEE Workshop on Signal Processing (Poznań, October 2003)* pp 7–14

[5] De Vos A and Van Rentergem Y 2005 Reversible computing: from mathematical group theory to electronical circuit experiment *Proc. Computing Frontiers Conference (Ischia, May 2005)* pp 35–44

[6] Feynman R 1985 Quantum mechanical computers *Opt. News* **11** 11–20

[7] Van Rentergem Y, De Vos A and Storme L 2005 Implementing an arbitrary reversible logic gate *J. Phys. A: Math. Gen.* **38** 3555–77

[8] De Vos A and Van Rentergem Y 2005 Synthesis of reversible circuits *Proc. 14th Int. Workshop on Logic and Synthesis (Lake Arrowhead, June 2005)* pp 101–8

[9] Kerber A 1971 Representations of Permutation groups: I *(Lecture Notes in Mathematics* vol 240*)* (Berlin: Springer)

[10] Maslov D, Dueck G and Miller D 2005 Toffoli network synthesis with templates *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **24** 807–17

[11] Maslov D, Dueck G and Miller D 2003 Fredkin/Toffoli templates for reversible logic synthesis *Proc. Int. Conf. on CAD (San Jose, November 2003)* pp 256–61

[12] Shende V, Prasad A, Markov I and Hayes J 2003 Synthesis of reversible logic circuits *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **22** 710–22

[13] De Vos A and Storme L 2004 r-Universal reversible logic gates *J. Phys. A: Math. Gen.* **37** 5815–24

[14] De Vos A, Raa B and Storme L 2002 Generating the group of reversible logic gates *J. Phys. A: Math. Gen.* **35** 7063–78

[15] Desoete B and De Vos A 2002 A reversible carry-look-ahead adder using control gates *Integr. VLSI J.* **33** 89–104