Journal of Universal Computer Science, vol. 5, no. 5 (1999), 307-321
submitted: 14/1/99, accepted: 14/4/99, appeared: 28/5/99 *© Springer Pub. Co.*

Group Theoretical Aspects of Reversible Logic Gates

Leo Storme
(Vakgroep zuivere wiskunde en computeralgebra,
Universiteit Gent, Belgium
Galglaan 2, B - 9000 Gent
ls@cage.rug.ac.be)

Alexis De Vos
(Imec v.z.w.,
Universiteit Gent, Belgium
Sint Pietersnieuwstraat 41, B - 9000 Gent
alex@elis.rug.ac.be)

Gerald Jacobs
(Vakgroep wiskundige natuurkunde en sterrenkunde,
Universiteit Gent, Belgium
Krijgslaan 281, B - 9000 Gent
gerald.jacobs@rug.ac.be)

Abstract: Logic gates with three input bits and three output bits have a privileged position within fundamental computer science: they are a sufficient building block for constructing arbitrary reversible boolean networks and therefore are the key to reversible digital computers. Such computers can, in principle, operate without heat production. As there exist as many as $8! = 40{,}320$ different 3-bit reversible truth tables, the question arises as to which ones to choose as building blocks. Because these gates form a group with respect to the operation 'cascading', we can apply group theoretical tools, in order to make such a choice.

Key Words: reversible computing, group theory, permutations

Category: B.6.1

1 Introduction

Conventional computers are built from basic building blocks, such as the `AND`, `NAND`, `OR`, `NOR`, and `XOR` gates. See Table 1. Such tables are logically irreversible. This means that, if we forget the value of the input (A, B), knowledge of the output P is not sufficient to calculate backwards and to recover the value of (A, B).

According to Landauer's principle [1] [2] [14] [15] [18], 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 room temperature, this package of heat is small, i.e. 2.9×10^{-21} joule, 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 gates of Table 1. On the contrary, it consists exclusively

Table 1: Classical truth tables: (a) `AND` , (b) `NAND` , (c) `OR` , (d) `NOR` , (e) `XOR`.

AB	P
0 0	0
0 1	0
1 0	0
1 1	1

(a)

AB	P
0 0	1
0 1	1
1 0	1
1 1	0

(b)

AB	P
0 0	0
0 1	1
1 0	1
1 1	1

(c)

AB	P
0 0	1
0 1	0
1 0	0
1 1	0

(d)

AB	P
0 0	0
0 1	1
1 0	1
1 1	0

(e)

of logically reversible building blocks. The number of output bits of a reversible logic gate necessarily equals its number of input bits. We will call this number the 'logic width' of the gate. Fredkin and Toffoli [10] [19] have demonstrated that three inputs and three outputs is necessary and sufficient in order to construct a reversible implementation of an arbitrary boolean function of a finite number of logic variables. Thus, from the fundamental point of view, reversible logic gates with a width equal to three have a privileged position.

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, B, C, ...) and the w-bit output (P, Q, R, ...). For convenience, all possible inputs, ranging from (0, 0, 0, ...) to (1, 1, 1, ...), are ordered 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. In particular, there are $8! = 40{,}320$ reversible gates with 3-bit width.

The present paper investigates which of these 40,320 gates fulfil the role of universal building block, and which fulfil this job more efficiently than the others. In order to tackle the problem, we will successively study the reversible gates with $w = 1$, $w = 2$, and $w = 3$.

2 Calculation with a single bit

There exist only four different truth tables with one bit input and one bit output. Two of them are logically irreversible: the resetter ($P = 0$) and the setter ($P = 1$). The two others are reversible: the follower ($P = A$) and the inverter ($P =$ `NOT` A). If, for example, we have 'forgotten' the value of A, knowledge of the value of the inverter's output P suffices to recover it.

Note that among the 1-bit reversible gates, the `NOT` gate is a 'generator'. This means we can make any reversible gate of width 1 by combining a finite number of this particular gate. Indeed, a follower can be fabricated by the sequence of two inverters. The opposite is not true: one cannot fabricate an inverter by cascading followers.

3 Calculation with two bits

There are $4^4 = 256$ different truth tables with two inputs (A, B) and two outputs (P, Q). Among them, only $4! = 24$ are reversible. However, some of these twenty-four truth tables fall apart into two separate 1-bit reversible tables. E.g. Table 2a decomposes into one follower $Q = A$ (Table 2b) and one inverter $P =$ `NOT` B (Table 2c). On the contrary, truth Table 3b is an example of a 2-bit reversible table that cannot be reduced to two separate 1-bit reversible tables, and therefore is called a true two-bit reversible gate. Among the 24 reversible 2-bit tables, only 16 are true 2-bit tables.

Table 2: Falling apart of a truth table.

AB	PQ
0 0	1 0
0 1	0 0
1 0	1 1
1 1	0 1

(a)

$=$

A	Q
0	0
1	1

(b)

&

B	P
0	1
1	0

(c)

All reversible true 2-bit gates can be fabricated from the same building block, combined with an inverter before and/or an inverter after. Indeed, Table 3b together with the inverter (Table 3a) forms a set of two building blocks with which we can synthetize an arbitrary reversible 2-bit gate. Truth Table 3b is called the `CONTROLLED NOT` by Feynman [8] [9]. Its logic operation looks like this:

$$P = A$$
$$Q = A \texttt{ XOR } B \ ,$$

where `XOR` is the abbreviation of the `EXCLUSIVE OR` function. The gate is the reversible form of the classical (irreversible) `XOR` gate. The latter is represented in Table 1e.

Figure 1 gives a representative example of a 2-bit reversible gate, realized by combining `NOT` and `CONTROLLED NOT` gates. Whereas output Q simply equals input B, output P can be described in three different ways:

$$P = \texttt{ NOT } (A \texttt{ XOR } B)$$
$$P = A \texttt{ XOR } (\texttt{ NOT } B)$$
$$P = (\texttt{ NOT } A) \texttt{ XOR } B \ .$$

Table 3: Feynman's truth tables: (a) NOT, (b) CONTROLLED NOT, (c) CONTROLLED CONTROLLED NOT.

A	P
0	1
1	0

(a)

AB	PQ
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

(b)

$A\,B\,C$	$P\,Q\,R$
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)

These three boolean expressions are identical, but lead to different physical realizations. We note, however, that these implementations not only make use of the 1-bit NOT function and the 2-bit CONTROLLED NOT function, but also of the 2-bit exchanger, i.e. the gate that interchanges two logic variables (realizing $P = B$ as well as $Q = A$). This is an example of a general property: the EXCHANGER, the NOT, and the CONTROLLED NOT form a natural 'generating set' for the twenty-four 2-bit reversible gates. More precisely, each reversible 2-bit gate can be synthetized by taking one or zero CONTROLLED NOTs and adding one or zero EXCHANGERs and one or zero NOTs to the left and to the right of it. Note that neither the EXCHANGER nor the NOT is a true 2-bit gate, but the CONTROLLED NOT is one. See Table 4.

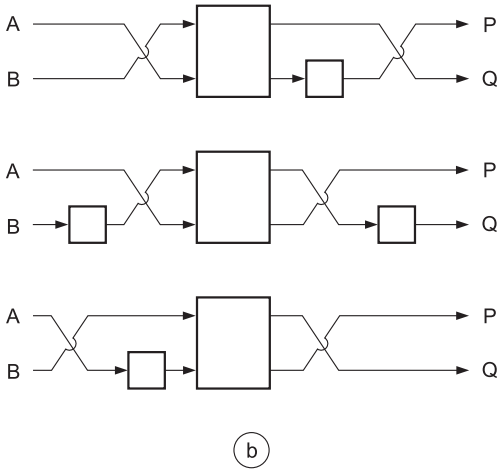
4 Calculation with three bits

There exist $8^8 = 16,777,216$ different truth tables with 3 inputs and 3 outputs. Among them, only $8! = 40,320$ are reversible. However, 48 of these truth tables fall apart into three separate 1-bit reversible tables and another 288 fall apart into one 1-bit and one (true) 2-bit reversible gate. Thus, among the 40,320 reversible 3-bit gates, only 39,984 are true 3-bit gates.

Two notorious examples are the Fredkin gate [10] [19] and Feynman's CONTROLLED CONTROLLED NOT gate [8] [9]. The truth table of the latter is given in Table 3c. The former is shown in Table 5a. Both have a particular property: each is a universal primitive. This means that any boolean function of any finite number of logic input variables can be implemented by combining a finite number of such building blocks. The proof consists of two steps [19]: one first proves that the building block suffices to implement the NAND function (Table 1b), then one refers to the fact that the NAND function is a universal primitive. The latter

AB	PQ
0 0	1 0
0 1	0 1
1 0	0 0
1 1	1 1

(a)



(b)

Figure 1: The synthesis of a 2-bit reversible gate: (a) truth table, (b) three different implementations combining NOTs and CONTROLLED NOT.

Table 4: The three generators of the 2-bit reversible gates: (a) EXCHANGER, (b) NOT, (c) CONTROLLED NOT.

AB	PQ
0 0	0 0
0 1	1 0
1 0	0 1
1 1	1 1

(a)

AB	PQ
0 0	0 1
0 1	0 0
1 0	1 1
1 1	1 0

(b)

AB	PQ
0 0	0 0
0 1	0 1
1 0	1 1
1 1	1 0

(c)

step is a well-known theorem. The former step is demonstrated by introducing a so-called preset: we keep one or two inputs fixed and look how the three outputs are function of the remaining input(s). Among the 39,984 reversible true 3-bit gates, many have the universality property. It is clear, however, that the number 39,984 is too large to allow 'manual' inspection. We have to recur to computer-algebra software specially dedicated to group theory, such as GAP [11] [17] and Magma [3] [12]. In the present study, we have chosen the GAP approach, because of GAP's built-in commands DoubleCoset and DoubleCosets.

Table 5: Truth tables: (a) Fredkin's conservative gate, (b) a 'pseudo-inverting' gate.

$A\,B\,C$	$P\,Q\,R$
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 0 1
1 1 0	1 1 0
1 1 1	1 1 1

`(2,3)`

(a)

$A\,B\,C$	$P\,Q\,R$
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	1 0 1
0 1 1	1 0 0
1 0 0	0 1 1
1 0 1	0 1 0
1 1 0	1 1 0
1 1 1	1 1 1

`(3,6)(4,5)`

(b)

5 Groups and subgroups

Because of the universality property of some of the 3-bit reversible gates, we now continue the $w = 3$ case in more detail.

When a reversible 3-bit gate x is cascaded by a reversible 3-bit gate y (i.e. when the P output of gate x is connected to the A input of y, etc.), then a new reversible 3-bit gate is formed, denoted xy. The 40,320 reversible truth tables of width 3 therefore form a group [16], say $\mathbf{R}$, which is isomorphic to the symmetric group $\mathbf{S_8}$. The identity element of the group is the 3-bit follower ($P = A$, $Q = B$, and $R = C$). In `GAP`, each element of $\mathbf{R}$ is denoted by its permutation notation. E.g. the follower is denoted `()`, whereas the `CONTROLLED CONTROLLED NOT` is written `(7,8)`, because the seventh and the eighth line of the truth table (i.e. Table 3c) are interchanged.

In order to classify the large number of elements of the group $\mathbf{R}$, we will introduce in the following paragraphs three different important subgroups, namely $\mathbf{E}$ with 6 elements, $\mathbf{F}$ with 48 elements, and finally $\mathbf{G}$ with 1,344 elements. They are ordered as

$$\mathbf{E} < \mathbf{F} < \mathbf{G} < \mathbf{R}\ ,$$

where $<$ denotes 'is proper subgroup of'. By means of each of these three subgroups, we will partition $\mathbf{R}$ into double cosets, which will serve as equivalence classes in the application of Section 6.

An important subgroup of $\mathbf{R}$ is formed by the follower together with the five elements representing mere relabellings. Table 6a shows the example e_1, satisfying $P = A$, $Q = C$, and $R = B$, i.e. performing an exchange of B and C. In

permutation notation, gate e_1 is written `(2,3)(6,7)`. A second example is e_2 or `(3,5)(4,6)`. See Table 6b. Together these two elements generate the whole subgroup of exchangers. The importance of this subgroup **E** comes from the fact that these gates are trivial to implement in any technology. E.g. in electronics, they consist merely of cross-overs of metal lines. The subgroup **E** of exchange gates contains six elements. It is denoted $G_6(SW)$ by Rayner and Newman [16] and is isomorphic to the symmetric group $\mathbf{S}_3$. Results from GAP show us that **E** partitions the full group **R** into 1,172 distinct double cosets. A double coset of an element g consists of all elements $e'ge''$, where both e' and e'' are elements of the subgroup **E**. This means that, although there exist 40,320 different reversible gates, there are only 1,172 'really different' ones, as soon as we consider exchangers as 'for free'. From these 1,172 gates, all other 39,148 gates can be fabricated by merely adding one relabelling gate to the left and one to the right.

In a next step, we can enlarge the above subgroup **E**, by introducing the inverter or NOT gate. One can either invert A (i.e. realize the gate $P =$ NOT A, $Q = B$, and $R = C$), or invert B (i.e. realize $P = A$, $Q =$ NOT B, and $R = C$), or invert C ($P = A$, $Q = B$, and $R =$ NOT C). As an example, the cycle notation of the last gate is `(1,2)(3,4)(5,6)(7,8)`. These three inverters (denoted i_1, i_2, and i_3) generate a subgroup of order $2^3 = 8$, isomorphic to $\mathbf{Z}_2^3$, where $\mathbf{Z}_2$ is the cyclic group of order 2. Together, the subgroup **E** of exchangers and the subgroup **I** of inverters generate a new subgroup **F** of order 48, isomorphic to $\mathbf{S}_3{:}\mathbf{Z}_2^3$, the semi-direct product of $\mathbf{S}_3$ and $\mathbf{Z}_2^3$. The elements of **F** are exactly the 48 gates mentioned in Section 4, i.e. the 3-bit gates that fall apart into three distinct 1-bit gates.

Using GAP, we find that the subgroup **F** partitions the full group **R** into 52 distinct double cosets. This means that, although there exist 40,320 different reversible gates, there are only 52 'really different' ones, if we consider both exchangers and inverters as 'for free'. From these 52 gates, corresponding to representatives of the 52 distinct double cosets of **F**, all other 40,268 gates can be fabricated by merely adding one free gate to the left and one to the right. Table 7 gives a list of all 52 double cosets k_i. Note that GAP gives them in a specific order, which has no a priori meaning for the user. We also get a representative r_i of class k_i. Again GAP's way of choosing this representative is not transparent to the user. The different double cosets constructed with **F** sometimes have different size. Table 7 gives n_i, i.e. the number of elements in the double coset. At first sight, it may be a surprise that a double coset may contain less than $48^2 = 2{,}304$ members. This is caused by the fact that different products $f'gf''$ (with g a member of **R** and both f' and f'' members of **F**) can lead to equal results. It is possible to prove that each double coset contains a number of elements which is a multiple of 48. Double coset k_1 is the subgroup **F** itself, with the follower `()` as representative r_1. We remark that Feynman's gate `(7,8)` is the representative r_4 of class k_4.

If we take the elements of subgroup **F** and add the representative r_i of k_i, then these 49 elements together generate a subgroup. Such a subgroup is called the closure of **F** and r_i. Its order we denote by m_i in Table 7. From GAP, we learn that

- Sometimes m_i is as large as 40,320, meaning that the closure of **F** and r_i is the full group **R**. In other words, any element of **R** can then be written as a finite product of form $f'r_if''r_if'''r_if''''...$, i.e. a finite cascade of r_i gates

separated by merely exchangers and/or inverters. In this case, we call r_i universal.
- Sometimes m_i is as small as n_i+48, meaning that k_i together with k_1 forms a subgroup. Any product of the form $f'r_if''r_if'''r_if''''...$ then generates either an element of k_1 or an element of k_i. The only double cosets with this property are k_3 and k_{31}.

In order to get more insight into the 52 double cosets in which **R** is divided, we construct the lattice of all subgroups containing **F** and contained in **R**. This yields a set of partially ordered subgroups: Figure 2. We note ten different subgroups:

- $k_1 = \mathbf{F}$
- $k_1 \cup k_3$ of order $192 = 4 \times 48$
- $k_1 \cup k_{31}$ of order $192 = 4 \times 48$
- $k_1 \cup k_2 \cup k_3$ of order $384 = 8 \times 48$
- $k_1 \cup k_{31} \cup k_{40}$ of order $576 = 12 \times 48$
- $k_1 \cup k_8 \cup k_{31} \cup k_{40} \cup k_{49}$ of order $1{,}152 = 24 \times 48$
- $k_1 \cup k_3 \cup k_{36} \cup k_{38}$ of order $1{,}344 = 28 \times 48$
- $k_1 \cup k_3 \cup k_{19} \cup k_{21} \cup k_{31} \cup k_{33} \cup k_{34}$ of order $1{,}344 = 28 \times 48$
- $k_1 \cup k_3 \cup k_5 \cup k_7 \cup k_9 \cup k_{11} \cup k_{13} \cup k_{15} \cup k_{18} \cup k_{19} \cup k_{21} \cup k_{23} \cup k_{25} \cup k_{28} \cup k_{31} \cup k_{33} \cup k_{34} \cup k_{36} \cup k_{38} \cup k_{40} \cup k_{41} \cup k_{43} \cup k_{45} \cup k_{46} \cup k_{48} \cup k_{50}$, forming the subgroup of all even permutations and thus isomorphic to the alternating group $\mathbf{A_8}$ of order 20,160
- the whole group **R** of order 40,320 itself.

Some of these subgroups have a particular interpretation. E.g. the subgroup $k_1 \cup k_8 \cup k_{31} \cup k_{40} \cup k_{49}$ is the closure of subgroup **F** and the subgroup of the 36 conservative gates. A conservative gate [10] is a gate where the output (P, Q, R) always has the same number of 1's as the input (A, B, C). Fredkin's gate `(2,3)` is an example. The subgroup $k_1 \cup k_2 \cup k_3$ is the closure of **F** and the subgroup of the 16 pseudo-inverting gates. We call a 'pseudo-inverting' gate a gate where the output (P, Q, R) always is equal to either the input (A, B, C) or to (`NOT` A, `NOT` B, `NOT` C). Its permutation notation consists merely of transpositions `(i,9-i)`. Table 5b shows an example. The meaning of the subgroup $k_1 \cup k_3 \cup k_{19} \cup k_{21} \cup k_{31} \cup k_{33} \cup k_{34}$ will become clear below.

In a final step, we can enlarge subgroup **F**, by adding a Feynman `CONTROLLED NOT`. See Table 6d. The resulting **G** is a subgroup of **R** and a supergroup of **F**. It is isomorphic to $\mathbf{2^3{:}L_3(2)}$, the semi-direct product of $\mathbf{2^3}$, i.e. the additive group of all binary vectors of length 3, and $\mathbf{L_3(2)}$, i.e. the multiplicative group of all non-singular binary 3×3 matrices. In detail, **G** is isomorphic to the multiplicative group of all non-singular binary 4×4 matrices of the form

$$\begin{pmatrix} & & & a_1 \\ & A & & a_2 \\ & & & a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with $\det(A) \neq 0$ and with $a_1, a_2, a_3 \in \{0, 1\}$. See Reference [4]. It is of order 1,344 and divides the full group **R** into four double cosets: see Table 8. Double coset K_1 is the subgroup **G** itself, represented by representative $R_1 =$ `()`. It is

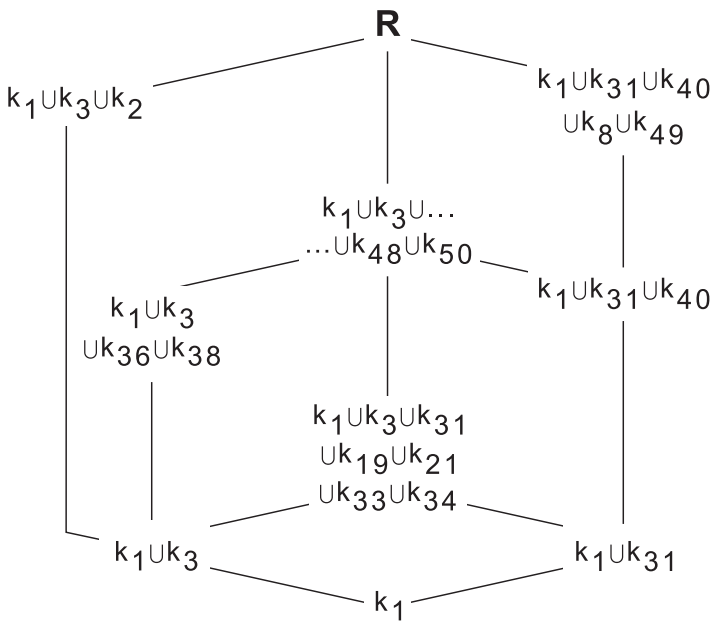


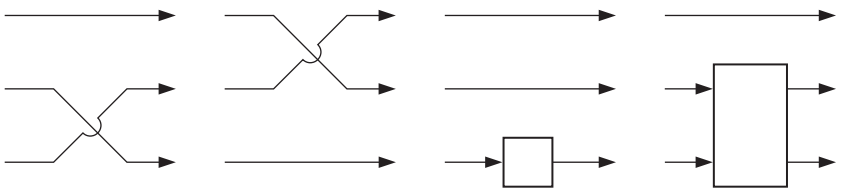
Figure 2: The lattice.

identical to the subgroup $k_1 \cup k_3 \cup k_{19} \cup k_{21} \cup k_{31} \cup k_{33} \cup k_{34}$ encountered above. It contains

- all 48 reversible 3-bit gates that fall apart into three distinct 1-bit gates,
- all 288 reversible 3-bit gates that fall apart into one 1-bit gate and one true 2-bit gate,
- another 1,008 gates, which are true 3-bit gates, but can be constructed by cascading two (or more) non-true 3-bit gates.

One can easily check that none of the 1,344 elements of **G** is universal and that all 38,976 members of the three other double cosets, i.e. K_2, K_3, and K_4 are universal.

Table 6: The subgroup generators: (a) `EXCHANGER`, (b) `EXCHANGER`, (c) `NOT`, (d) `CONTROLLED NOT`. From top to bottom, four different representations: schematic, set of logic equations, truth table, and permutation. Gates (a) and (b) together generate subgroup **E**; Gates (a), (b), and (c) together generate subgroup **F**; Gates (a), (b), (c), and (d) together generate subgroup **G**.



(a)	(b)	(c)	(d)
$P = A$ $Q = C$ $R = B$	$P = B$ $Q = A$ $R = C$	$P = A$ $Q = B$ $R =$ `NOT` C	$P = A$ $Q = B$ $R = B$ `XOR` C
$A\,B\,C$ | $P\,Q\,R$ 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	$A\,B\,C$ | $P\,Q\,R$ 0 0 0 | 0 0 0 0 0 1 | 0 0 1 0 1 0 | 1 0 0 0 1 1 | 1 0 1 1 0 0 | 0 1 0 1 0 1 | 0 1 1 1 1 0 | 1 1 0 1 1 1 | 1 1 1	$A\,B\,C$ | $P\,Q\,R$ 0 0 0 | 0 0 1 0 0 1 | 0 0 0 0 1 0 | 0 1 1 0 1 1 | 0 1 0 1 0 0 | 1 0 1 1 0 1 | 1 0 0 1 1 0 | 1 1 1 1 1 1 | 1 1 0	$A\,B\,C$ | $P\,Q\,R$ 0 0 0 | 0 0 0 0 0 1 | 0 0 1 0 1 0 | 0 1 1 0 1 1 | 0 1 0 1 0 0 | 1 0 0 1 0 1 | 1 0 1 1 1 0 | 1 1 1 1 1 1 | 1 1 0
`(2,3)(6,7)`	`(3,5)(4,6)`	`(1,2)(3,4)` `(5,6)(7,8)`	`(3,4)(7,8)`

Table 7: The double cosets k_i of **F** in **R**.

i	n_i	m_i	r_i
1	$48 = 48 \times 1$	$48 = 48 \times 1$	()
2	$192 = 48 \times 4$	$384 = 48 \times 8$	(4,5)
3	$144 = 48 \times 3$	$192 = 48 \times 4$	(3,4)(5,6)
4	$576 = 48 \times 12$	$40320 = 48 \times 840$	(7,8)
5	$576 = 48 \times 12$	$20160 = 48 \times 420$	(4,5)(7,8)
6	$576 = 48 \times 12$	$40320 = 48 \times 840$	(2,3,4)(5,8,7,6)
7	$576 = 48 \times 12$	$20160 = 48 \times 420$	(2,3,5,8,7,6,4)
8	$288 = 48 \times 6$	$1152 = 48 \times 24$	(6,7)
9	$576 = 48 \times 12$	$20160 = 48 \times 420$	(4,5)(6,7)
10	$576 = 48 \times 12$	$40320 = 48 \times 840$	(3,4)(5,7,6)
11	$576 = 48 \times 12$	$20160 = 48 \times 420$	(3,5,7,6,4)
12	$288 = 48 \times 6$	$40320 = 48 \times 840$	(2,5,7,4,3,6)
13	$2304 = 48 \times 48$	$20160 = 48 \times 420$	(6,7,8)
14	$2304 = 48 \times 48$	$40320 = 48 \times 840$	(4,5)(6,7,8)
15	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(2,3,4)(5,8,6)
16	$1152 = 48 \times 24$	$40320 = 48 \times 840$	(2,3,5,8,6,4)
17	$1152 = 48 \times 24$	$40320 = 48 \times 840$	(2,4)(3,7,5,8,6)
18	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(2,5,8,6,3,7,4)
19	$288 = 48 \times 6$	$1344 = 48 \times 28$	(5,6)(7,8)
20	$576 = 48 \times 12$	$40320 = 48 \times 840$	(4,6,5)(7,8)
21	$288 = 48 \times 6$	$1344 = 48 \times 28$	(2,3,4)(5,8,7)
22	$1152 = 48 \times 24$	$40320 = 48 \times 840$	(5,6,7,8)
23	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(4,6,7,8,5)
24	$576 = 48 \times 12$	$40320 = 48 \times 840$	(3,4)(5,7,8)
25	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(3,6,4)(5,7,8)
26	$576 = 48 \times 12$	$40320 = 48 \times 840$	(2,6,3,7,4)(5,8)
27	$288 = 48 \times 6$	$40320 = 48 \times 840$	(5,6,8,7)
28	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(4,6,8,7,5)
29	$576 = 48 \times 12$	$40320 = 48 \times 840$	(3,4)(5,8,7)
30	$288 = 48 \times 6$	$40320 = 48 \times 840$	(2,5,8,7)(3,6,4)
31	$144 = 48 \times 3$	$192 = 48 \times 4$	(5,8)(6,7)
32	$576 = 48 \times 12$	$40320 = 48 \times 840$	(4,8,5)(6,7)
33	$288 = 48 \times 6$	$1344 = 48 \times 28$	(3,4)(5,7,6,8)
34	$144 = 48 \times 3$	$1344 = 48 \times 28$	(2,6,3,7)(5,8)
35	$576 = 48 \times 12$	$40320 = 48 \times 840$	(4,5,6)(7,8)
36	$576 = 48 \times 12$	$1344 = 48 \times 28$	(4,6)(7,8)
37	$576 = 48 \times 12$	$40320 = 48 \times 840$	(2,3,5,8,7,4)
38	$576 = 48 \times 12$	$1344 = 48 \times 28$	(2,3,6,5,8,7,4)
39	$1152 = 48 \times 24$	$40320 = 48 \times 840$	(4,5,6,7)
40	$384 = 48 \times 8$	$576 = 48 \times 12$	(4,6,7)

Table 7: The double cosets k_i of **F** in **R** (continued).

i	n_i	m_i	r_i
41	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(3,6,5,7,4)
42	$384 = 48 \times 8$	$40320 = 48 \times 840$	(2,4,3,6,5,7)
43	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(4,5,6,7,8)
44	$2304 = 48 \times 48$	$40320 = 48 \times 840$	(4,6,7,8)
45	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(3,5,7,8,4)
46	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(2,3,5,8,4)
47	$2304 = 48 \times 48$	$40320 = 48 \times 840$	(2,3,6,5,8,4)
48	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(2,6,3,7,5,8,4)
49	$288 = 48 \times 6$	$1152 = 48 \times 24$	(4,6,7)(5,8)
50	$1152 = 48 \times 24$	$20160 = 48 \times 420$	(4,8,5,6,7)
51	$576 = 48 \times 12$	$40320 = 48 \times 840$	(3,6,8,5,7,4)
52	$288 = 48 \times 6$	$40320 = 48 \times 840$	(2,3,4,8,5,6)

6 Application

Which of **R**'s three subgroups (**E**, **F**, or **G**) is of importance, depends on the technological circumstances. In almost each technology the exchangers are 'for free'. E.g. in electronics, they are implemented by a mere metal cross-over. In the special case of dual-line electronics, each logic variable is represented by two metal lines of opposite logic value, e.g. A together with `NOT` A. Therefore, in dual-rail electronics, also an inverter is 'free of charge': we only need a metal cross-over to exchange A with `NOT` A. In this technology, the `CONTROLLED NOT` is not for free. Thus we are in the case of the free subgroup **F** [5] [6] [7].

The following question arises [13]. We like to be able to synthetize any arbitrary member of **R** with the help of a limited number of generators. In electronics, we say: we like to implement any arbitrary reversible 3-bit gate with the help of a library with a limited number of standard cells. If we denote by s_1, s_2, ..., s_m the different members of the library, then an arbitrary member r of **R** must satisfy

$$r = f' s' f'' s'' f''' ... s^{(n)} f^{(n+1)} \ , \tag{1}$$

where s', s'', ..., $s^{(n)}$ are elements of the library $\{s_1, s_2, ..., s_m\}$ and f', f'', f''', ..., $f^{(n+1)}$ are elements of $\mathbf{F} = \{f_1, f_2, ..., f_{48}\}$. The number n is called the 'logic depth' of the implementation. In order to minimize the number of standard cells, we have to choose them from different double cosets and not from double coset k_1. Thus the library will be a subset of the representatives r_2, r_3, ..., r_{52}. From Table 7, it follows that a library with a single building block is sufficient: each of the representatives r_4, r_6, r_{10}, r_{12}, r_{14}, r_{16}, r_{17}, r_{20}, r_{22}, r_{24}, r_{26}, r_{27}, r_{29}, r_{30}, r_{32}, r_{35}, r_{37}, r_{39}, r_{42}, r_{44}, r_{47}, r_{51}, and r_{52} have $m_i = 40,320$ and are thus sufficient to generate the whole group **R**. But, these 23 solutions are not

equivalent. Indeed, we not only want to limit the number p of different building blocks in the library, we also like to limit the number of times we have to use the blocks, i.e. we like to minimize the depth n of the products (1). It turns out that with building block r_{14} all elements of $\mathbf{R}$ can be synthetized with $n \leq 4$. The elements of k_1 need no r_{14} block (i.e. $n = 0$); the elements of k_{14} need only one r_{14} block (i.e. $n = 1$); most elements of $\mathbf{R}$ need $n = 2$ or $n = 3$; only the elements of k_{34} need four cascaded r_{14} blocks ($n = 4$); on the average an arbitrary class of $\mathbf{R}$ needs $32/13 = 2.46$ building blocks in cascade. Exactly the same results apply to the building block r_{44} and to r_{47}. Table 9 compares these three optimal choices to the other twenty, i.e. less efficient, choices. In particular, we see in the 18 th line that Feynman's gate r_4 needs $0 \leq n \leq 6$ (with expectation value $97/26 = 3.73$) in order to generate all $\mathbf{R}$.

None of the double cosets k_{14}, k_{44}, and k_{47} appears in the lattice of Figure 2, except at the top, in the parent group $\mathbf{R}$ itself. Indeed, any element of any subgroup of Figure 2 can only generate other elements of that particular subgroup. The underlying reason is clear: such elements show 'too much symmetry'. E.g. conservative gates (such as Fredkin's gate) can, by cascading, only generate elements of $k_1 \cup k_8 \cup k_{31} \cup k_{40} \cup k_{49}$, such that no finite depth n can generate the other elements of $\mathbf{R}$. Finally, it is remarkable that the optimum double cosets k_{14}, k_{44}, and k_{47} are among the largest double cosets of Table 7: $n_{14} = n_{44} = n_{47} = 2,304$.

If we consider depth $n = 4$ as too deep a cascade (too much silicon surface area), we can construct a larger library. If we choose an $p = 2$ library, there are four equivalent optimal combinations: r_{14} together with r_{18}, r_{14} together with r_{41}, r_{44} together with r_{48}, and r_{44} together with r_{50}. Now we have $n \leq 3$, with expectation value $101/52 = 1.94$. Enlarging the library to $p = 3$ yields $n \leq 2$ and average cascade depth $99/52 = 1.90$.

Table 8: The double cosets K_i of $\mathbf{G}$ in $\mathbf{R}$.

i	N_i	M_i	R_i
1	$1344 = 1344 \times 1$	$1344 = 1344 \times 1$	()
2	$9408 = 1344 \times 7$	$40320 = 1344 \times 30$	(7,8)
3	$18816 = 1344 \times 14$	$20160 = 1344 \times 15$	(6,7,8)
4	$10752 = 1344 \times 8$	$40320 = 1344 \times 30$	(4,5)(6,7,8)

7 Conclusion

The reversible gates of width w form a group, isomorphic to the symmetric group $\mathbf{S}_{2^w}$. Group theory in general, and double cosets in particular, are well

Table 9: Cascade depth n.

k_i	$n_{\max}$	n_{ave}
k_{14}	4	2.46
k_{44}	4	2.46
k_{47}	4	2.46
k_{22}	4	2.88
k_{17}	4	2.92
k_{39}	4	2.92
k_{16}	5	3.44
k_{6}	5	3.54
k_{10}	5	3.54
k_{20}	5	3.54
k_{35}	5	3.54
k_{24}	5	3.58
k_{26}	5	3.58
k_{29}	5	3.58
k_{32}	5	3.58
k_{37}	5	3.58
k_{51}	5	3.58
k_{4}	6	3.73
k_{30}	6	4.23
k_{52}	6	4.23
k_{42}	7	4.35
k_{27}	7	4.77
k_{12}	8	5.71

suited to detect different classes within the $(2^w)!$ elements of the group. This can lead to an optimized choice of a set of generators. In electronics, this means an optimal set of hardware building blocks. With the help of `GAP`, we identified optimal gates g that are able to generate all other elements of $\mathbf{R}$ by means of a product of the form $f'gf''gf'''...$ of minimal length.

Acknowledgement

Leo Storme is research associate of the Fund for Scientific Research – Flanders (Belgium).

References

[1] Bennett, C.: "Logical reversibility of computation"; *I.B.M. Journal of Research and Development* **17** (1973), 525-532.

[2] Bennett, C., Landauer, R.: "The fundamental physical limits of computation"; *Scientific American* **253** (July 1985), 38-46.

[3] Bosma, W., Cannon, J., Playoust, C.: "`The Magma Algebra System I`: the user language"; *Journal of Symbolic Computation* **3-4** (1997), 235-265.

[4] Conway, J.H., Curtis, R.T., Norton, S.P., Parker, R.A., Wilson, R.A.: "Atlas of finite groups"; Oxford University Press, New York (1985), p. 22.

[5] De Vos, A.: "Introduction to r-MOS systems"; *Proc. 4 th Workshop on Physics and Computation*, Boston (1996), 92-96.

[6] De Vos, A.: "Towards reversible digital computers"; *Proc. European Conference on Circuit Theory and Design*, Budapest (1997), 923-931.

[7] De Vos, A.: "Reversible computing"; *Progress in Quantum Electronics* **23** (1999), 1-49.

[8] Feynman, R.: "Quantum mechanical computers"; *Optics News* **11** (1985), 11-20.

[9] Feynman, R.: "Feynman lectures on computation" (A. Hey and R. Allen, eds); Addison-Wesley, Reading (1996).

[10] Fredkin, E., Toffoli, T.: "Conservative logic"; *International Journal of Theoretical Physics* **21** (1982), 219-253.

[11] `http://www.can.nl/SystemsOverview/Special/GroupTheory/GAP/index.html`

[12] `http://www.maths.usyd.edu.au:8000/u/magma/index.html`

[13] Jacobs, G.: "Algebra der reversibele logische schakelingen"; M.Sc. thesis, Universiteit Gent, Gent (1998).

[14] Keyes, R., Landauer, R.: "Minimal energy dissipation in logic"; *I.B.M. Journal of Research and Development* **14** (1970), 153-157.

[15] Landauer, R.: "Irreversibility and heat generation in the computational process"; *I.B.M. Journal of Research and Development* **5** (1961), 183-191.

[16] Rayner, M., Newman, D.: "On the symmetry of logic"; *Journal of Physics A: Mathematical and General* **28** (1995), 5623-5631.

[17] Schönert, M.: "`GAP`"; *Computer Algebra Nederland Nieuwsbrief* **9** (1992), 19-28.

[18] Stix, G.: "Riding the back of electrons"; *Scientific American* **279** (September 1998), 20-21.

[19] Toffoli, T.: "Reversible computing"; in: "Automata, languages and programming" (J. De Bakker and J. Van Leeuwen, eds); Springer, New York (1980), pp. 632–644.