Reversible computing : from mathematical group theory to electronical circuit experiment

by Alexis De Vos and Yvan Van Rentergem
Imec v.z.w. and vakgroep elektronica en informatiesystemen
Universiteit Gent
Sint Pietersnieuwstraat 41
B - 9000 Gent
Belgium

December 8, 2004

Abstract

Reversible logic gates of a certain logic width w form a group (isomorphic to the symmetric group of order $(2^w)!$). Study of the subgroups of this group both learns us a lot about properties of reversible gates and guides us to synthesize particular circuits. After design, circuits are implemented in prototype silicon chips.

1 Introduction

Reversible computing [1] is useful both in lossless classical computing [2] and in quantum computing [3]. They can be implemented in both classical and quantum hardware technologies. In the present paper, we will illustrate its implementation into classical MOS electronics. We will demonstrate the application of group theory to the detailed design.

2 Groups

A group G consists of two components:

- a non-empty set S and
- an operation Ω.

The operation is binary, i.e. is a function of two elements of the set. The cardinal number of the set S is called the order of the group G. However, a set and an operation are not sufficient in order to have a group. Set and operation have to fulfil some conditions:

- Ω has to be associative: $(a\ \Omega\ b)\ \Omega\ c = a\ \Omega\ (b\ \Omega\ c)$, for all subsets $\{a, b, c\}$ of S;
- S has to be closed: $a\ \Omega\ b \in S$ for each subset $\{a, b\}$ of S;
- S has to have an identity element i: $a\ \Omega\ i = a$ for each element a of S;
- each element of S has to have an inverse element a^{-1} in S: $a\ \Omega\ a^{-1} = i$.

In the following, the set S consists of logic gates and the binary operation Ω is the cascading of two logic gates. It is clear that the set of all possible logic gates does **not** form a group. Table 1 shows an example. The truth Table 1a describes a logic gate with two binary inputs (A and B) and three binary outputs (P, Q, and R). The attentive reader will remark the purpose of this gate: it calculates the `OR`, the `AND`, and the `XOR` of the inputs: $P = A$ `OR` B, $Q = A$ `AND` B, and $R = A$ `XOR` B, or, in short-hand notation: $P = A + B$, $Q = AB$, and $R = A \oplus B$. If we denote this gate by a, then there clearly exists a gate i, such that $a\ \Omega\ i = a$. Indeed, Table 1b shows the truth table of such identity gate. However, the reader will vainly search for the inverse of a: there exists no gate b, such that $a\ \Omega\ b$ equals i.

Reversible logic gates distinguish themselves from arbitrary logic gates by two properties:

- the number of output bits equals the number of input bits and
- for each pair of different input words, the two corresponding output words are different.

Table 2a gives an example r. The number of inputs equals the number of outputs, i.e. two. This number is called the width w of the reversible gate. The table gives all possible input words AB. We see how all the corresponding output words PQ are different. Therefore, in contrast to arbitrary logic gates, reversible logic gates **do** form a group. Table 2b gives the identity gate i and Table 2c gives r^{-1}, i.e. the inverse of r. The reader will easily verify that not only the cascade $r\ \Omega\ r^{-1}$, but also the cascade $r^{-1}\ \Omega\ r$ equals i. The truth Table 2c of r^{-1} is deduced from the truth Table 2a of r by reading backwards: for each word AB in Table 2c, we look for the identical string PQ in Table 2a and we copy the corresponding AB

Table 1: Truth table of two logic gates: (a) an arbitrary gate a, (b) the identity gate i.

AB	PQR
0 0	0 0 0
0 1	1 0 1
1 0	1 0 1
1 1	1 1 0

(a)

ABC	PQR
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

(b)

Table 2: Truth table of three reversible logic gates of width 2: (a) an arbitrary reversible gate r, (b) the identity gate i, and (c) the inverse r^{-1} of r.

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

(a)

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

(b)

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

(c)

word from Table 2a in the PQ column of Table 2c. The fact that, in the truth table of reversible gates, all output words PQ are different, guarantees that this algorithm actually works.

Because group theory is not applicable in the case of arbitrary boolean logic, it is no surprise that group theory is not popular among designers of combinatorial logic. As a historical legacy, also designers of reversible logic circuits hardly use group theory. This really is a pity, as group theorists provide us with a wealth of insights, theorems, and tools. It would be a terrible waste not to apply these. In the present paper, we will only demonstrate a few applications of group theory for the understanding and synthesis of reversible logic networks. We will profitably make use of subgroups, cosets, and double cosets.

3 Subgroups

All reversible gates of same width form a group. If we denote by w the width, then the truth table of an arbitrary reversible gate has 2^w rows. As all output words have to be different, they can merely be a repetition of the input words in a different order. In other words: the 2^w output words are a permutation of the 2^w input words. There exist $(2^w)!$ ways to permute 2^w objects. Therefore there exist exactly $(2^w)!$ different reversible logic gates of width w. Thus $(2^w)!$ is the order of the group. The group is isomorphic to a group well-known by mathematicians: the symmetric group $\mathbf{S}_{2^w}$.

The symmetric group has a wealth of properties. It e.g. has a lot of subgroups, of which most have been studied in detail. A set S of m elements has $2^m - 1$ different subsets (the trivial subset S included, but the trivial empty subset $\emptyset$ not included). A group G of order g has fewer than $2^g - 1$ different subgroups. Indeed, Lagrange's Theorem tells us that the order h of a subgroup H of G has to be a divisor of the order g of its supergroup G. Thus $\mathbf{S}_{2^w}$ of order $(2^w)!$ has less than $2^{(2^w)!} - 1$ subgroups. In spite of this, the group $\mathbf{S}_{2^w}$ still has very many subgroups, as soon as $w > 2$. See the Appendix. Some of these subgroups naturally make their appearance in the study of reversible computing. An example is the subgroup of conservative logic gates. Conservative reversible gates have been studied in detail by Fredkin and Toffoli [4]. A reversible gate is called conservative, if the weight of the output word $PQR...$ always equals the weight of the corresponding input word $ABC....$ Here, the weight of a word is defined as the number of 1s that appear in it. It is e.g. clear that the gate r of Table 2a is not conservative, as $AB = 10$ of weight 1 gives rise to $PQ = 11$ of weight 2.

Table 3: Truth table of three reversible logic gates of width 3: (a) a conservative gate, (b) a linear gate, (c) a selective gate, and (d) an exchanging gate.

ABC	PQR
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	1 1 0
1 1 0	0 1 1
1 1 1	1 1 1

(a)

ABC	PQR
0 0 0	1 0 0
0 0 1	0 0 0
0 1 0	0 0 1
0 1 1	1 0 1
1 0 0	1 1 1
1 0 1	0 1 1
1 1 0	0 1 0
1 1 1	1 1 0

(b)

ABC	PQR
0 0 0	1 0 0
0 0 1	1 0 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	0 1 0
1 1 1	0 1 1

(c)

ABC	PQR
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

(d)

On the contrary, Table 3a is a conservative gate of width 3. It is clear that, among the 8! permutations of the 8 objects 000, 001, 010, 011, 100, 101, 110, and 111, only combinations of

- the 3! permutations of the 3 objects 001, 010, and 100 and
- the 3! permutations of the 3 objects 011, 101, and 110

are allowed. Whereas there are as many as $8! = 40,320$ reversible gates of width 3, there are only $(3!)^2 = 36$ conservative gates of width 3. In general: the group of conservative gates of width w is isomorphic to the group $\mathbf{S}_{C_w^1} \times \mathbf{S}_{C_w^2} \times ... \times \mathbf{S}_{C_w^{w-1}}$ (direct product of different symmetric groups) and is of order $(C_w^1)!\ (C_w^2)!\ ...\ (C_w^{w-1})!$. It is a subgroup of the group of reversible gates of order $(2^w)!$.

An even more important subgroup is the subgroup of linear reversible gates. Linear reversible gates have been studied in detail by Patel et al. [5]. A logic gate is linear iff each of its outputs P, Q, ... is a linear function of the inputs A, B, ... In its turn, a linear function is defined as follows. A boolean function $f(A, B, ...)$ is linear iff its Reed–Muller expansion only contains terms of degree 0 and terms of degree 1. The reversible gate of Table 3a is not linear. Indeed it can be written as a set of three boolean

functions:

$$\begin{aligned} P &= B \oplus AB \oplus AC \\ Q &= A \\ R &= C \oplus AB \oplus AC \ . \end{aligned}$$

Whereas the function $Q(A,B,C)$ is linear, the function $P(A,B,C)$ is clearly not (its Reed–Muller expansion containing two terms of second degree). Table 3b is an example of a linear gate:

$$\begin{aligned} P &= 1 \oplus B \oplus C \\ Q &= A \\ R &= A \oplus B \ . \end{aligned}$$

Why are linear gates so important? Because of a non-property. Based on pioneering work by Kerntopf [6], De Vos and Storme [7] have proved that an arbitrary boolean function can be synthesized by a (loop-free and fanout-free) wiring of a finite number of identical reversible gates, provided that gate is not linear. In other words: all non-linear reversible gates can be used as a universal building block. Thus the linear reversible gates constitute the 'weak' ones. Indeed, any wiring of linear gates (be they reversible or not, be they identical or not) can only yield linear boolean functions at its outputs. The linear reversible gates form a group isomorphic to what is called in mathematics the affine linear group $ALG(w,2)$. Its order equals $2^{(w+1)w/2}\, w!_2$, where $w!_2$ is the bifactorial of w, the q-factorial being a generalization of the ordinary factorial $w! = w!_1$:

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

Table 4 gives the number of different linear reversible gates. We see that (at least for $w > 2$), a vast majority of reversible gates are non-linear and thus universal. This is in remarkable constrast to conventional boolean algebra, where most logic gates (e.g. `NOT`, `AND`, `OR`, `XOR`, and `NXOR` gates) lack the universality property. Being a universal building-block (e.g. `NAND` and `NOR` gates) is considered there as a 'special status'. In irreversible logic, only a minority is universal. In the reversible world, it turns out to be the other way around: the 'strong ones' form the majority. E.g., for $w = 3$, not less that 97 % of the reversible gates can be used as a universal primitive. For $w = 4$, even 99.999998 % are universal.

Now we can go a step further: does a finite number of copies of a single arbitrary linear reversible gate suffice to synthesize an arbitrary given linear

Table 4: The number r of different reversible gates, the number l of different linear reversible gates, the number s of different selective reversible gates, and the number e of different exchanging reversible gates, as a function of the gate width w.

w	r	l	s	e
1	2	2	2	1
2	24	24	8	2
3	40,320	1,344	48	6
4	20,922,789,888,000	322,560	384	24

boolean function? The reader will easily verify that this is not true. In order to be able to synthesize any linear boolean function, it is necessary and sufficient that the linear building block should have at least one output that contains at least two different degree-one terms in its Reed–Muller expansion. Table 3b is such gate, as e.g. the Reed–Muller expansion of P contains both the terms B and C (see above). Reversible Table 3c, on the contrary, is linear, but not linear-universal, as each of its outputs is only function of one input:

$$\begin{aligned} P &= 1 \oplus B \\ Q &= A \\ R &= C \ . \end{aligned}$$

Such gates are called selective reversible gates. They form a subgroup of order $w!2^w$. This number is given in Table 4. The group is isomorphic to the indirect product $\mathbf{S}_w{:}\mathbf{S}_2^w$ of the symmetric group $\mathbf{S}_w$ with the wth power of the symmetric group $\mathbf{S}_2$ (which, by the way, is isomorphic to the cyclic group $\mathbf{Z}_2$ of order two).

If we replace the condition 'each of its outputs is only function of one input' by 'each of its outputs equals one input', we again descend the hierarchy of subgroups. Table 3c is such a gate:

$$\begin{aligned} P &= B \\ Q &= A \\ R &= C \ . \end{aligned}$$

Such gates are called exchangers. They form a subgroup isomorphic to $\mathbf{S}_w$ of order $w!$. Also this number is given in Table 4. Finally, we can impose 'each of the outputs equals the corresponding input':

$$\begin{aligned} P &= A \\ Q &= B \\ R &= C \ . \end{aligned}$$

This results in the trivial subgroup $\mathbf{I}$ of order 1, merely consisting of the identity gate i.

We have thus constructed a chain of subgroups:

$$\mathbf{S}_{2^w} \supset ALG(w,2) \supset \mathbf{S}_w{:}\mathbf{Z}_2^w \supset \mathbf{S}_w \supset \mathbf{I} \quad ,$$

with subsequent orders

$$(2^w)! > 2^{(w+1)w/2}\, w!_2 > w!2^w > w! > 1 \ .$$

Here, the symbol $\supset$ reads 'is proper supergroup of'. For the example $w = 3$, this becomes:

$$\mathbf{S}_8 \supset ALG(3,2) \supset \mathbf{S}_3{:}\mathbf{Z}_2^3 \supset \mathbf{S}_3 \supset \mathbf{I} \quad ,$$

with subsequent orders

$$40,320 > 1,344 > 48 > 6 > 1 \ .$$

Note that we have ordered here only a small sample of the many subgroups of $\mathbf{S}_8$. If we want to describe the relationship between all subgroups, only a partial ordering is possible. Instead of a chain of subgroups, we then have a grid of subgroups. See e.g. reference [8].

4 Cosets

Subgroups are at the origin of a second powerful tool in group theory, i.e. cosets. If $\mathbf{H}$ (with order h) is a subgroup of the group $\mathbf{G}$ (with order g), then $\mathbf{H}$ partitions $\mathbf{G}$ into $\frac{g}{h}$ classes, all of same size h. These equipartition classes are called cosets. We distinguish left cosets and right cosets.

The left coset of the element a of $\mathbf{G}$ is defined as all elements of $\mathbf{G}$ which can be written as a cascade $b\ \Omega\ a$, where b is an arbitrary element of $\mathbf{H}$. Such left coset forms an equipartition class, because of the following property: if c is member of the left coset of a, then a is member of the left coset of c.

Right cosets are defined in an analogous way: the right coset of the element a of $\mathbf{G}$ is defined as all elements of $\mathbf{G}$ which can be written as a cascade $a\ \Omega\ b$, where b is an arbitrary element of $\mathbf{H}$. Note that $\mathbf{H}$ itself is one of the left cosets of $\mathbf{G}$, as well as one of its right cosets. Note that all other cosets are not subgroups of $\mathbf{G}$, as they lack the identity element i. That is the reason why we talk about cosets, not about cogroups.

What is the reason of defining cosets? They are very handy in synthesis. Assume we want to make an arbitrary element of the group $\mathbf{G}$ in hardware. Instead of solving this problem for each of the g cases, we only synthesize the h gates b of $\mathbf{H}$ and a single representative r_i of each other left coset ($1 \leq i \leq \frac{g}{h} - 1$). If we can make each of these $h + \frac{g}{h} - 1$ gates, we can make all the others by merely making a short cascade $b\ \Omega\ r_i$. If we cleverly choose the subgroup $\mathbf{H}$, we can guarantee that $h + \frac{g}{h} - 1$ is much smaller that g. We call the set of $h + \frac{g}{h} - 1$ building-blocks the library for synthesizing the g gates of $\mathbf{G}$. What is a clever choice of $\mathbf{H}$? The library will be as small as possible if $\frac{d}{dh}\ (h + \frac{g}{h} - 1)$ is zero, i.e. if $h = \sqrt{g}$. As Lagranges Theorem tells us that only h-values can exist which are divisor of g, we thus have to choose a subgroup $\mathbf{H}$ with order 'close to' the square root of the order of $\mathbf{G}$.

Maslov and Dueck [9] present a method for synthesizing an arbitrary reversible gate of width three. As a subgroup $\mathbf{H}$ of the group $\mathbf{S}_8$, they propose all gates with output PQR equal 000 in case of the input $ABC =$ 000. This subgroup is isomorphic to $\mathbf{S}_7$. Thus the supergroup has order $g = 40,320$, whereas the subgroup has order $h = 5,040$. The subgroup divides the supergroup into 8 cosets. Because h is quite larger than $\sqrt{g} \approx$ 201, this choice is not optimal, but still $h + \frac{g}{h} - 1 = 5,047$ is much smaller that 40,320. Interesting is the fact, that the procedure can be repeated: for designing each of the 5,040 members of $\mathbf{S}_7$, Maslov and Dueck choose a subgroup of $\mathbf{S}_7$. They choose all reversible gates where PQR equals 000 in case $ABC = 000$ and equals 001 in case $ABC = 001$. This is a subgroup isomorphic to $\mathbf{S}_6$ of order 6! = 720, which divides $\mathbf{S}_7$ into seven cosets. Ectetera. Thus Maslov and Dueck apply the following chain of subgroups:

$$\mathbf{S}_8 \supset \mathbf{S}_7 \supset \mathbf{S}_6 \supset \mathbf{S}_5 \supset \mathbf{S}_4 \supset \mathbf{S}_3 \supset \mathbf{S}_2 \supset \mathbf{S}_1 = \mathbf{I}\ .$$

with subsequent orders;

$$40,320 > 5,040 > 720 > 120 > 24 > 6 > 2 > 1\ .$$

We can conclude that, for synthesizing all 40,320 members of $\mathbf{S}_8$, we need a library of only $7 + 6 + 5 + 3 + 2 + 1 = 28$ elements. Together with the identity gate, they suffice to synthesize an arbitrary member of $\mathbf{S}_8$ by a

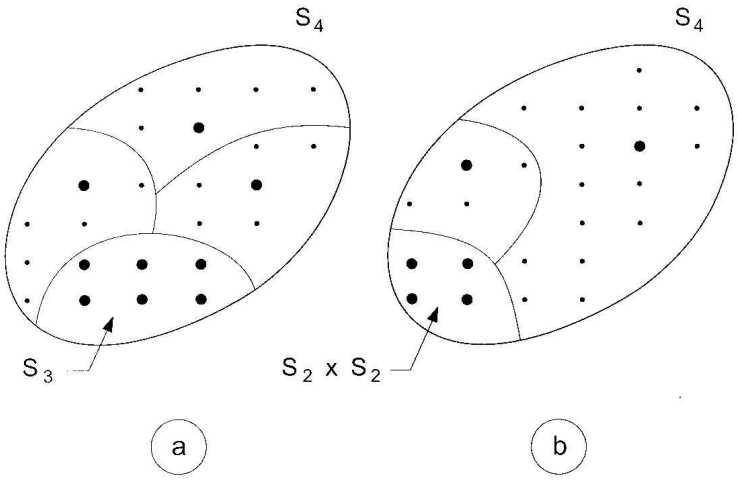

Figure 1: The symmetric group $\mathbf{S}_4$ partitioned (a) as the four left cosets of $\mathbf{S}_3$ and (b) as the three double cosets of $\mathbf{S}_2 \times \mathbf{S}_2$.
Note: the dots depict the elements of $\mathbf{S}_4$; the bold-faced dots depict the elements of the subgroup and the representatives of the (double) cosets.

cascade with length of seven or less. Figure 1a illustrates one step of the procedure: the 24 elements of $\mathbf{S}_4$ are fabricated by means of the 6 elements of its the subgroup $\mathbf{S}_3$ plus the representatives of the 3 other cosets in which $\mathbf{S}_4$ is partitioned by $\mathbf{S}_3$.

Besides Maslov and Dueck, also Shende et al. [10], Agrawal and Jha [11] and Kerntopf [12] [13] have presented synthesis methods that are based on subgroups and cosets. However, whereas the Maslov–Dueck method is straightforward, their methods are partially heuristic. Finally, note that none of these authors is aware they are using subgroups and cosets.

5 Double cosets

Even more powerful than cosets are double cosets. The double coset of a, element of $\mathbf{G}$, is defined as the set of all elements that can be written as $b_1\ \Omega\ a\ \Omega\ b_2$, where both b_1 and b_2 are members of the subgroup $\mathbf{H}$. A surprising fact is that, in general, the double cosets, in which $\mathbf{G}$ is divided by $\mathbf{H}$, are of different sizes (ranging from h to h^2). The number of double cosets, in which $\mathbf{G}$ is divided by $\mathbf{H}$, therefore is not easy to recover. It is some number between $1 + \frac{g-h}{h^2}$ and $\frac{g}{h}$. Usually, the number is much smaller

than $\frac{g}{h}$, leading to the (appreciated) fact that there are far less double cosets than there are cosets.

For the problem of synthesizing all members of $\mathbf{S}_8$, Van Rentergem and De Vos [14] have chosen the double cosets of the following subgroup: among the 8! permutations of the 8 objects 000, 001, 010, 011, 100, 101, 110, and 111, only combinations of

- the 4! permutations of the 4 objects 000, 001, 010, and 011 and
- the 4! permutations of the 4 objects 100, 101, 110, and 111

are allowed. This subgroup has the property $P = A$, is isomorphic to $\mathbf{S}_4^2$ and has order $(4!)^2 = 576$. Subsequently, the members of $\mathbf{S}_4$ are divided into double cosets by use of its subgroup $\mathbf{S}_2^2$, etcetera. Thus, finally, Van Rentergem and De Vos apply the following chain of subgroups:

$$\mathbf{S}_8 \supset \mathbf{S}_4^2 \supset \mathbf{S}_2^4 \supset \mathbf{S}_1^8 = \mathbf{I} \quad .$$

with subsequent orders

$$40,320 > 576 > 16 > 1 \ .$$

We can conclude that, for synthesizing all 40,320 members of $\mathbf{S}_8$, they need a library of only $4 + 2 + 1 = 7$ elements. Together with the identity gate, these suffice to synthesize an arbitrary member of $\mathbf{S}_8$ by a cascade with length of seven or less. Figure 1b illustrates one step of the procedure: the 24 elements of $\mathbf{S}_4$ are fabricated by means of the 4 elements of its subgroup $\mathbf{S}_2 \times \mathbf{S}_2$ plus the representatives of the 2 other double cosets in which $\mathbf{S}_4$ is partitioned by $\mathbf{S}_2 \times \mathbf{S}_2$.

6 Electronics

Whether we use left cosets, or right cosets, or double cosets in the synthesis procedure, whether we choose one subgroup or another, does not make such a big difference. Finally, we have a procedure for synthesizing an arbitrary gate by cascading a small number of standard cells from a limited library. By appropriate choice of the representatives of the (double) cosets, we can see to it that all building blocks in the library are member of either of the two following special subgroups:

- the subgroup of exchangers and

- the subgroup of simple [1] control gates.

The former group has order $w!$ and is already discussed in Section 3. The latter group is new. It consists of reversible gates, with w inputs A, B, C, ..., J, and K and w outputs P, Q, R, ..., Y, and Z, such that each output equals its corresponding input, except the last one:

$$\begin{aligned} P &= A \\ Q &= B \\ R &= C \\ ... &= ... \\ Y &= J \\ Z &= f(A,B,C,...,J) \oplus K \ , \end{aligned}$$

where f is an arbitrary boolean function of the $w-1$ variables A, B, C, ..., J. The reader will easily verify that, whatever function f we choose, the gate is reversible. As there exist 2^{2^q} different functions of q boolean variables, the group consists of $2^{2^{w-1}}$ elements. Among them we note three special elements:

- If f is identically zero, then Z is always equal to K. Then the gate is the identity gate i.
- If f is identically one, then Z always equals $1 \oplus K$. Then the gate is an inverter or `NOT` gate: $Z = \overline{K}$.
- If $f(A,B,C,...,J)$ equals the $(w-1)$-bit `AND` function $ABC...J$, then the gate is called the `CONTROLLED`$^{w-1}$ `NOT` gate or `TOFFOLI` gate. The former name is explained as follows: whenever $ABC...J$ equals 0, then Z simply equals K; but whenever $ABC...J$ equals 1, then Z equals `NOT` K.

The subgroup of simple control gates has been studied in detail by De Vos et al. [15].

For physical implementation, dual logic is very convenient. It means that any logic variable X is represented by two physical quantities, the first representing X itself, the other representing `NOT` X. Thus, e.g. the physical

[1] The adjective 'simple' comes from the fact that reference [15] defines, apart from the subgroup of 'simple control gates' (order $2^{2^{w-1}}$) also a larger group of 'control gates' (order 2^{2^w-1}). For sake of simplicity, in the present paper, we do not discuss this latter group.

gate realizing logic gate of Table 2a has four physical inputs: A, `NOT` A, B, and `NOT` B, or, in short-hand notation: A, $\overline{A}$, B, and $\overline{B}$. It also has four physical outputs: P, $\overline{P}$, Q, and $\overline{Q}$. Such approach is common in electronics, where it is called dual-line or dual-rail electronics. Also some quantum computers make use of dual-rail qubits [16]. As a result, half of the input pins are at logic 0 and the other half at logic 1, and analogous for the output pins. In this way, dual electronics is physically conservative: the number of 1s at the output equals the number of 1s at the input (i.e. equals w), even if the truth table of the logic gate is not conservative. As a result, we get the advantages of conservative logic, without having to restrict ourselves to conservative logic.

Dual-line hardware allows very simple implementation of the inverter. It suffices to interchange its two physical lines in order to invert a variable, i.e. in order to hardwire the `NOT` gate. Conditional `NOT`s are `NOT` gates which are controlled by switches. A first example is the `CONTROLLED NOT` gate:

$$\begin{aligned} P &= A \\ Q &= A \oplus B \ . \end{aligned}$$

These logic relationships are implemented into physical world as follows:

- output P is simply connected to input A,
- output $\overline{P}$ is simply connected to input $\overline{A}$,
- output Q is connected to input B if $A = 0$,
  but connected to $\overline{B}$ if $A = 1$, and
- output $\overline{Q}$ is connected to input $\overline{B}$ if $A = 0$,
  but connected to B if $A = 1$.

The last two implementations are shown in Figure 2a. In the figure, the arrow heads show the position of the switches if the accompanying label is 1. A second example is the `CONTROLLED CONTROLLED NOT` gate or `TOFFOLI` gate:

$$\begin{aligned} P &= A \\ Q &= B \\ Q &= AB \oplus C \ . \end{aligned}$$

Its logic relationships are implemented into physical world as follows:

- output P is simply connected to input A,

- output $\overline{P}$ is simply connected to input $\overline{A}$,
- output Q is simply connected to input B,
- output $\overline{Q}$ is simply connected to input $\overline{B}$,
- output R is connected to input C if either $A = 0$ or $B = 0$, but connected to $\overline{C}$ if both $A = 1$ and $B = 1$, and
- output $\overline{R}$ is connected to input $\overline{C}$ if either $A = 0$ or $B = 0$, but connected to C if both $A = 1$ and $B = 1$.

The last two implementations are shown in Figure 2b. Note that in both Figures 2a and 2b, switches always appear in pairs, of which one is closed whenever the other is open and vice versa. Such pair of switches has been called one 'Y-branch switch' by Forsberg [17]. It is clear that the above design philosophy can be extrapolated to a simple control gate with arbitrary control function f. Suffice it to wire a square circuit like in Figures 2a and 2b, with the appropriate series and parallel connection of switches.

Now that we have an implementation approach, we can realize any reversible circuit in hardware. We will demonstrate here some examples of implementation into electronic chip. In electronic circuits, a switch is realized by the use of a so-called transmission gate, i.e. two MOS-transistors in parallel (one n-MOS transistor and one p-MOS transistor). As an example [18], Figure 3 shows a 4-bit ripple adder, implemented in 2.4 μm standard c-MOS technology, consisting of eight `CONTROLLED NOT`s and eight `CONTROLLED CONTROLLED NOT`s, and thus of a total of 192 transistors. This prototype chip was fabricated in 1998. A second example [19] (Figure 4) was fabricated in 2000, in submicron technology: a 4-bit carry-look-ahead adder, implemented in 0.8 μm standard c-MOS technology, containing four `CONTROLLED NOT`s, four control gates of width $w = 3$, and one complex control gate of width $w = 13$. It contains a total of 320 transistors.

Switches not only can decide whether an input variable is inverted or not. We can apply switches also in order to decide whether two input variables are swapped or not. This concept leads to the `CONTROLLED SWAP` gate or `FREDKIN` gate [4]:

$$\begin{aligned} P &= A \\ Q &= B \oplus AB \oplus AC \\ R &= C \oplus AB \oplus AC \ . \end{aligned}$$

Figure 2c shows the physical implementation. The reader will easily extrapolate the design philosophy to reversible logic gates of width $w = w_1 + w_2$,

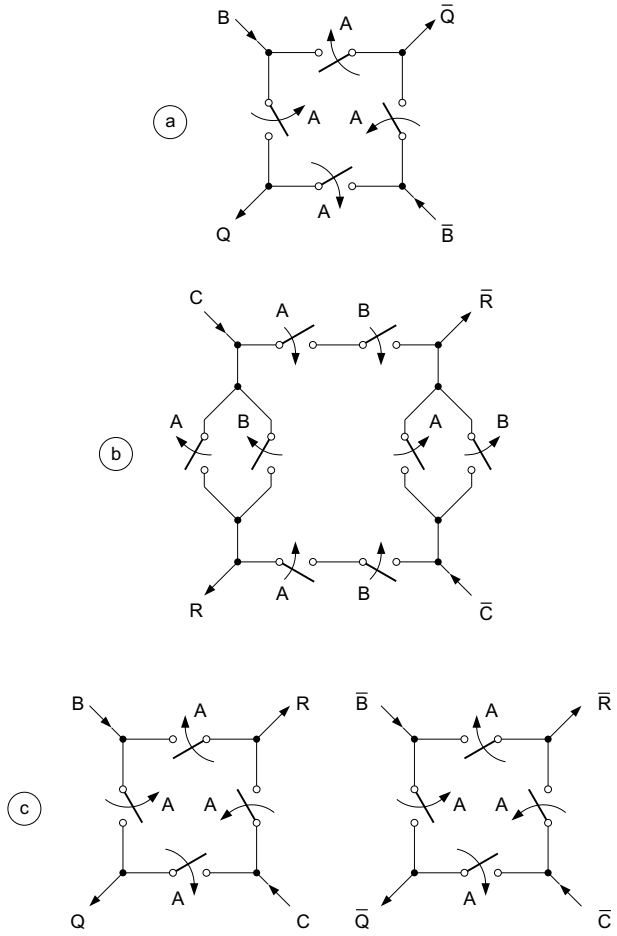

Figure 2: Schematic for (a) `CONTROLLED NOT` gate, (b) `CONTROLLED CONTROLLED NOT` gate, and (c) `CONTROLLED SWAP` gate.

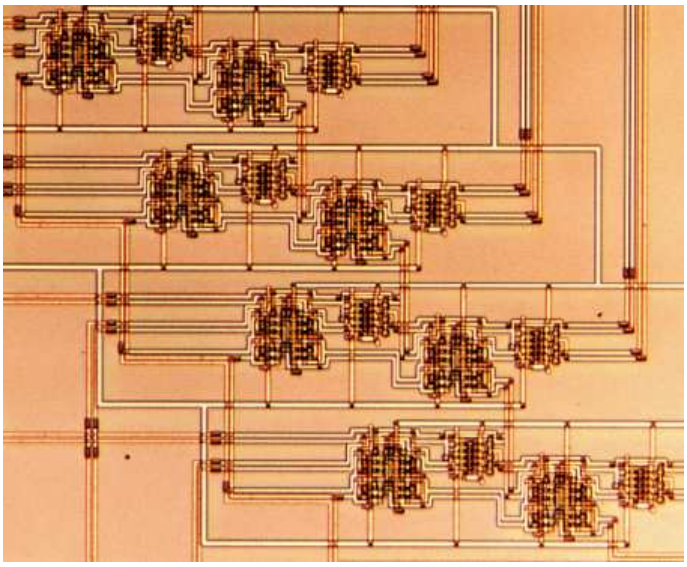

Figure 3: Microscope photograph (140 μm $\times$ 120 μm) of 2.4-μm 4-bit reversible ripple adder.

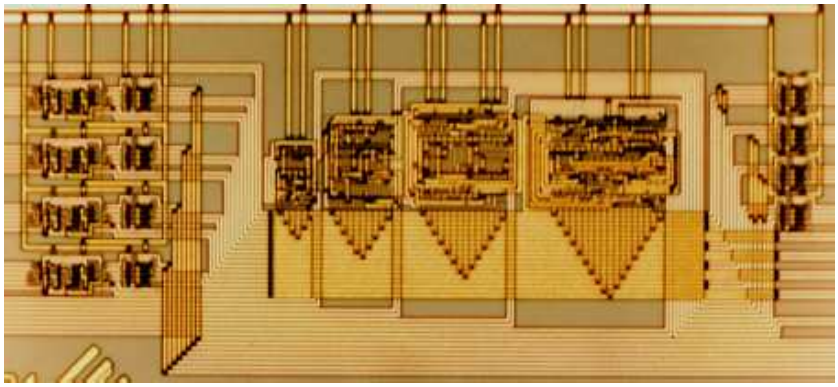

Figure 4: Microscope photograph (610 μm $\times$ 290 μm) of 0.8-μm 4-bit reversible carry-look-ahead adder.

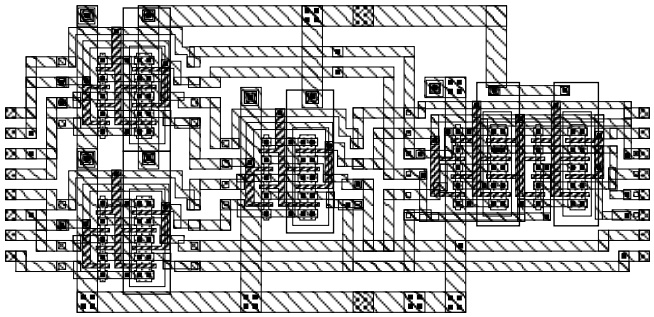

Figure 5: Computer layout (65 μm $\times$ 30 μm) of 0.35-μm 1-bit reversible full adder.

where w_1 controlling bits decide, by means of a control function f, whether the w_2 controlled bits are subjected to a given selective reversible gate or not. As there exist $2^{2^{w_1}}$ possible control functions and $2^{w_2} w_2!$ possible selective gates, we can construct a library of $2^{2^{w_1}} \times 2^{w_2} w_2!$ such building blocks.

An application [20] (Figure 5) is a 1-bit (full) adder, implemented in 0.35 μm standard c-MOS technology, containing three `CONTROLLED NOT`s and one `FREDKIN` gate. It contains a total of 40 transistors. The prototype chip was fabricated in 2004. The reader will observe that full-custom prototyping at university follows Moore's law, with a couple of years delay with respect to industry. Indeed, many commercial chips nowadays use 0.18 or 0.13 μm transistors. Some companies even have entered the nanoscale era, by introducing 90 nm products (and smaller) to the market.

The continuing shrinking of the transistor sizes leads to a continuing decrease of the energy dissipation per computational step. This heat generation Q is of the order of magnitude of CV_t^2, where V_t is the threshold voltage of the transistors and C is the total capacitance of the capacitors in the logic gate [21]. We have C of the order of magnitude of $\epsilon_0 \epsilon \frac{WL}{t}$, where W, L, and t are the width, the length, and the oxide thickness of the transistors, whereas ϵ_0 is the permittivity of vacuum (8.85×10^{-12} F/m) and ϵ is the dielectric constant of the gate oxide. Table 5 gives some typical numbers. The dielectric constant is taken 3.9 (silicondioxide SO_2). We see how Q becomes smaller and smaller. However, dissipation in electronic circuits still is about four orders of magnitude in excess of the Landauer quantum $kT \log(2)$, which amounts (for $T = 300$ K) to about 3×10^{-21} J or 3 zeptojoule.

Table 5: Moore's law for dimensions W, L, and t, and for threshold voltage V_t, as well as for resulting capacitance C and heat dissipation Q.

technology	W (μm)	L (μm)	t (nm)	V_t (V)	C (fF)	Q (fJ)
2.4	2.4	2.4	42.5	0.9	46.8	38
0.8	0.8	2.0	15.5	0.75	3.6	2.0
0.35	0.35	0.5	7.4	0.6	0.82	0.30

Further shrinking of W and L and further reduction of V_t ultimately will lead to a Q value in the neighboorhood of $kT\log(2)$. That day, digital electronics will have good reason to be reversible... This, however, does not mean that the reversible MOS circuits are useless today. Indeed, as they are a reversible form of pass-transistor topology, called r-MOS [22], they are particularly suited for adiabatic addressing [23], leading to substantial power saving. Figure 6 shows an example of a quasi-adiabatic experiment. We see two transient signals: one of the input variables and one of the resulting output bits. In practice, such procedure leads to a factor of about 10 in power reduction [21]. The reduction of the power dissipation is even more impressive if standard c-MOS technology is replaced by SOI (silicon-on-insulator) technology. Indeed, in the latter process, the threshold voltage V_t can be controlled better, such that low-V_t technologies are possible.

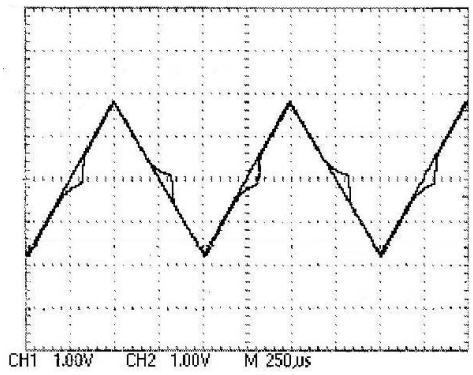

Figure 6: Oscilloscope view of 0.35 μm full adder.

7 Conclusion

We have demonstrated that subgroup chains of the form $\mathbf{S}_8 \supset ALG(3,2) \supset \mathbf{S}_3 : \mathbf{Z}_2^3 \supset \mathbf{S}_3 \supset \mathbf{S}_1$ divide the group of reversible logic gates in sets of elements that are less and less suited as building-blocks for circuit design. We have shown how cosets and double cosets are particularly helpful for synthesizing, in a systematic way, arbitrary reversible circuits. In particular, we have demonstrated the power of subgroup chains of the form $\mathbf{S}_8 \supset \mathbf{S}_7 \supset \mathbf{S}_6 \supset \mathbf{S}_5 \supset \mathbf{S}_4 \supset \mathbf{S}_3 \supset \mathbf{S}_2 \supset \mathbf{S}_1$ and the form $\mathbf{S}_8 \supset \mathbf{S}_4^2 \supset \mathbf{S}_2^4 \supset \mathbf{S}_1^8$. Finally, we have presented, as an example, the design and prototyping of three different adders in three different c-MOS technologies.

References

[1] I. Markov : " An introduction to reversible circuits ", *Proceedings of the International Workshop on Logic and Synthesis*, Laguna Beach (May 2003), pp. 318 - 319.

[2] A. De Vos : " Lossless computing ", *Proceedings of the I.E.E.E. Workshop on Signal Processing*, Poznań (October 2003), pp. 7 - 14.

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

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

[5] K. Patel, I. Markov, and J. Hayes : " Optimal synthesis of linear reversible circuits ", *Proceedings of the 13 th International Workshop on Logic and Synthesis*, Temecula (June 2004), pp. 470 - 477.

[6] P. Kerntopf : " On universality of binary reversible logic gates ", *Proceedings of the 5 th Workshop on Boolean Problems*, Freiberg (September 2002), pp. 47 - 52.

[7] A. De Vos and L. Storme : " r-Universal reversible logic gates ", *Journal of Physics A: Mathematical and General* **37** (2004), pp. 5815 - 5824.

[8] L. Storme, A. De Vos, and G. Jacobs : " Group theoretical aspects of reversible logic gates ", *Journal of Universal Computer Science* **5** (1999), pp. 307 - 321.

[9] D. Maslov and G. Dueck : " Reversible cascades with minimal garbage ", *I.E.E.E. Transactions on Computer-Aided Design of Integrated Circuits and Systems* **23** (2004), pp. 1497 - 1509.

[10] V. Shende, A. Prasad, I. Markov, and J. Hayes : " Synthesis of reversible logic circuits ", *I.E.E.E. Transactions on Computer-Aided Design of Integrated Circuits and Systems* **22** (2003), pp. 710 - 722.

[11] A. Agrawal and N. Jha : " Synthesis of reversible logic ", *Proceedings of the Design, Automation and Test in Europe Conference*, Paris (February 2004), pp. 1384 - 1385.

[12] P. Kerntopf : " A new heuristic algorithm for reversible logic circuit synthesis ", *Proceedings of the 41 th Design Automation Conference*, San Diego (June 2004), pp. 834 - 837.

[13] P. Kerntopf : " Reversible logic circuit synthesis based on a new complexity measure ", *Proceedings of the 13 th International Workshop on Logic and Synthesis*, Temecula (July 2004), pp. 106 - 113.

[14] Y. Van Rentergem, A. De Vos, and L. Storme : " Implementing an arbitrary reversible logic gate ", *Journal of Physics A: Mathematical and General*, to be published.

[15] A. De Vos, B. Raa, and L. Storme : " Generating the group of reversible logic gates ", *Journal of Physics A : Mathematical and General* **35** (2002), pp. 7063 - 7078.

[16] I. Chuang and Y. Yamamoto : " The dual-rail quantum bit and quantum error correction ", *Proceedings of the 4 th Workshop on Physics and Computation*, Boston (November 1996), pp. 82 - 91.

[17] E. Forsberg : " Reversible logic based on electron waveguide Y-branch switches ", *Nanotechnology* **15** (2004), pp. S298 - S302.

[18] B. Desoete, A. De Vos, M. Sibiński, and T. Widerski : " Feynman's reversible logic gates, implemented in silicon ", *Proceedings of the 6 th International Conference on Mixed Design of Integrated Circuits and Systems*, Kraków (June 1999), pp. 497 - 502.

[19] B. Desoete and A. De Vos : " A reversible carry-look-ahead adder using control gates ", *Integration, the V.L.S.I. Journal* **33** (2002), pp. 89 - 104.

[20] Y. Van Rentergem and A. De Vos : " Optimal design of a reversible full adder ", *International Journal of Unconventional Computing*, to be published.

[21] A. De Vos and Y. Van Rentergem : " Energy dissipation in reversible logic addressed by a ramp voltage ", *Proceedings of the 15 th International PATMOS Workshop*, Leuven (September 2005), to be published.

[22] A. De Vos : " Introduction to r-MOS systems ", *Proceedings of the 4 th Workshop on Physics and Computation*, Boston (November 1996), pp. 92 - 96.

[23] P. Patra and D. Fussell : " On efficient adiabatic design of MOS circuits ", *Proceedings of the 4 th Workshop on Physics and Computation*, Boston (November 1996), pp. 260 - 269.

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

[25] R. Singer : " Group Unit : permutation groups ", `http://www.fractions-plus.com/Ab%20Alg/Permutation%20Groups.doc` (2003).

[26] N. Sloane, S. Plouffe, and A. Hulpke : " The on-line encyclopedia of integer sequences ", `http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A005432` (2004).

Appendix : how many subgroups ?

How many subgroups does the symmetric group $\mathbf{S}_n$ have? There is no straightforward answer to this. We give here the example of $\mathbf{S}_4$ of order $4! = 24$. As a set of 24 elements has $2^{24} - 1 = 16,777,215$ subsets, we know that the number of subgroups is smaller. By brute force method, the computer algebra package `GAP` [24] gives the answer:

- There is the trivial subgroup of order 24, isomorphic to $\mathbf{S}_4$.
- There is one subgroup of order 12, isomorphic to the alternating group $\mathbf{A}_4$.
- There are 3 conjugate subgroups of order 8, all Sylow 2-subgroups isomorphic to $\mathbf{Z}_2^3$.

- There are 4 conjugate subgroups of order 6, isomorphic to $\mathbf{S}_3$.
- There are 7 subgroups of order 4, of which
  - three are isomorphic to $\mathbf{Z}_4$ and
  - four are isomorphic to $\mathbf{Z}_2^2$.
- There are 4 conjugate subgroups of order 3, all Sylow 3-subgroups isomorphic to $\mathbf{Z}_3$.
- There are 9 subgroups of order 2, all isomorphic to $\mathbf{Z}_2$.
- There is one (trivial) subgroup of order 1.

This gives a total of 30 subgroups. The reader will easily verify that, in accordance with Lagranges Theorem, all subgroups have an order which is a divisor of the order of $\mathbf{S}_4$. The reader will also verify that the reverse property holds. If a number is a divisor of the order of the group $\mathbf{S}_4$, then there exists at least one subgroup of order equal to that particular divisor. However the reverse property is not generally true. E.g. the order of $\mathbf{S}_5$ is 120, a number which has sixteen divisors: $\{1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 24, 30, 40, 60, 120\}$. There exist no subgroups of $\mathbf{S}_5$ with order equal to either 15, or 30, or 40. See e.g. reference [25].

Table 6 gives, for n up to 10, the order g of the group $\mathbf{S}_n$ and the number s of its subgroups [26]. We see how the number of subgroups is indeed substantially smaller than the number of subsets. For $n = 4$, we have $30 \ll 2^{24}-1 \approx 2\times10^{7}$; for $n = 8$, we have $151,221 \ll 2^{40,320}-1 \approx 3\times10^{12,137}$.

Acknowledgement.

The authors thank the *Invomec* division of *Imec v.z.w.* (Leuven, Belgium) and the *Eurochip/Europractice* organization, for processing the chips at *Mietec Alcatel Microelectronics* (2.4 μm, Oudenaarde, Belgium), at *Austria Mikro Systeme* (0.8 μm, Unterpremstätten, Austria), and at *AMI Semiconductor* (0.35 μm, Oudenaarde, Belgium).

Table 6: The order g of the symmetric group $\mathbf{S}_n$ and the number s of subgroups of $\mathbf{S}_n$. 'Important' cases (i.e. where n is a power of 2) are boldfaced.

n	g	s
1	1	1
2	**2**	**2**
3	6	6
4	**24**	**30**
5	120	156
6	720	1,455
7	5,040	11,300
8	**40,320**	**151,221**
9	362,880	1,694,723
10	3,628,800	29,594,446