Int. Journ. of Unconventional Computing, Vol. 4, pp. 79–88
Reprints available directly from the publisher
Photocopying permitted by license only

From Group Theory to Reversible Computers

ALEXIS DE VOS, AND YVAN VAN RENTERGEM

Imec v.z.w. and Universiteit Gent, B-9000 Gent, Belgium
E-mail: Alex.DeVos@elis.UGent.be

Received: December 1, 2006. Accepted: December 19, 2006.

Reversible logic circuits of a certain logic width form a group, isomorphic to a symmetric group. Its Young subgroups allow systematic synthesis of an arbitrary reversible circuit. We can choose either a left coset, right coset, or double coset approach. The tools are beneficial to both classical and quantum computers.

Keywords: reversible computing, group theory, young subgroup

1 INTRODUCTION

Reversible computing [1] is useful both in lossless classical computing [2] [3] and in quantum computing [4]. It can be implemented in both classical and quantum hardware technologies. In the present paper, we will demonstrate the application of group theory to the detailed design.

Reversible logic circuits distinguish themselves from arbitrary logic circuits by two properties: (1) the number of output bits always equals the number of input bits and (2) for each pair of different input words, the two corresponding output words are different. For instance, it is clear that an AND gate is not reversible, as (a) it has only one output bit, but two input bits and (b) for three different input words, the output words are equal. Table 1, on the other hand, gives two examples of a reversible circuit. Here, the number of inputs equals the number of outputs, i.e. three. This number is called the width w of the reversible circuit. The table gives all possible input words ABC. We see how all the corresponding output words PQR are different.

TABLE 1
Truth table of two reversible logic circuits of width 3: (a) a linear circuit and (b) the `MILLER` gate.

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

(a)

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

(b)

2 GROUP THEORY

All reversible circuits of the same width form a group. If we denote by w the width, then the truth table of an arbitrary reversible circuit 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 circuits of width w. The number 2^w is called the degree of the group; the number $(2^w)!$ is called the order of the group. The group is isomorphic to a group well-known by mathematicians: the symmetric group $\mathbf{S}_{2^w}$. For further reading in the area of symmetric groups in particular, and groups, subgroups, cosets, and double cosets in general, the reader is refered to appropriate textbooks [5].

The symmetric group has a wealth of properties. For example, it has a lot of subgroups, of which most have been studied in detail. Some of these subgroups naturally make their appearance in the study of reversible computing. An important subgroup is the subgroup of linear reversible circuits, studied in detail by Patel et al. [6]. A logic circuit is linear iff each of its outputs P, Q, ... is a linear function of the inputs A, B, ... A Boolean function $f(A, B, ...)$ is called linear iff its Reed–Muller expansion [7] only contains terms of degree 0 and terms of degree 1. The reversible circuit of Table 1a is not linear. Indeed it can be written as a set of three Boolean equations:

$$P = B \oplus AB \oplus AC$$
$$Q = A$$
$$R = C \oplus AB \oplus AC \ .$$

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 1b is an example of a linear circuit:

$$P = 1 \oplus B \oplus C$$
$$Q = A$$
$$R = A \oplus B \ .$$

De Vos and Storme [8] 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 the gate is not linear. In other words: all non-linear reversible circuits can be used as a universal building block.

3 COSETS

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

The left coset of the element a of **G** is defined as all elements of **G** which can be written as a cascade ba, where b is an arbitrary element of **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. Note that **H** itself is one of the left cosets of **G**, as well as one of its right cosets.

What is the reason of defining cosets? They are very handy in synthesis. Assume we want to make an arbitrary element of the group **G** in hardware. Instead of solving this problem for each of the g cases, we only synthesize the h circuits b of **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 a short cascade br_i. If we cleverly choose the subgroup **H**, then $h + \frac{g}{h} - 1$ is much smaller that g. We call the set of $h + \frac{g}{h} - 1$ building-blocks the library for synthesizing the g circuits of **G**.

Maslov and Dueck [9] present a method for synthesizing an arbitrary reversible circuit of width three. As subgroup **H** of the group **G** = $\mathbf{S}_8$, they propose all circuits with output PQR equal to 000 in case of the input $ABC = 000$. This subgroup is isomorphic to $\mathbf{S}_7$. Thus $g = 8! = 40,320$ and $h = 7! = 5,040$. The subgroup partitions the supergroup into 8 cosets. Interesting is the fact, that the procedure is 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 a subgroup isomorphic to $\mathbf{S}_6$ of order 6! = 720, which partitions $\mathbf{S}_7$ into seven cosets. Etcetera. 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 subgroup $\mathbf{S}_3$ plus the representatives of the 3 other cosets in which $\mathbf{S}_4$ is partitioned by $\mathbf{S}_3$. Thus Maslov and Dueck apply the following chain of

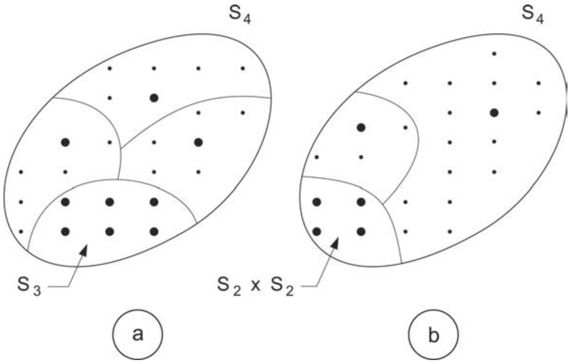

FIGURE 1
The symmetric group $\mathbf{S}_4$ partitioned (a) as the four left cosets of $\mathbf{S}_3$ and (b) as the three double cosets of $\mathbf{S}_2 \times \mathbf{S}_2$.
Note: the dots depict the elements of $\mathbf{S}_4$; the bold-faced dots depict the elements of the subgroup and the representatives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

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

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

Here, the symbol $\supset$ reads 'is proper supergroup of'. Maslov and Dueck need, for synthesizing all 40,320 members of $\mathbf{S}_8$, a library of only $(7+6+...+1)+1=29$ elements (the identity gate included).

Van Rentergem et al. [10] also present a coset synthesis method, however based on the following subgroup $\mathbf{H}$: all circuits from $\mathbf{G} = \mathbf{S}_8$ possessing the property $P = A$. It is isomorphic to $\mathbf{S}_4 \times \ \mathbf{S}_4 = \mathbf{S}_4^2$ and has order $h = (4!)^2 = 576$. The subgroup $\mathbf{S}_4^2$ partitions its supergroup $\mathbf{S}_8$ into 70 cosets. Subsequently, the members of $\mathbf{S}_4$ are partitioned into 144 cosets by use of its subgroup $\mathbf{S}_2^2$, etcetera. Thus, Van Rentergem et al. apply the following chain of subgroups:

$$\mathbf{S}_8 \supset \mathbf{S}_4^2 \supset \mathbf{S}_2^4 \supset \mathbf{S}_1^8 = \mathbf{I} \ , \qquad (1)$$

with subsequent orders

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

We note that the group $\mathbf{S}_{a-1}$ as well as the group $\mathbf{S}_{\frac{a}{2}} \times \ \mathbf{S}_{\frac{a}{2}}$ are special cases of Young subgroups of $\mathbf{S}_a$. In general, a Young subgroup [11] of the symmetric group $\mathbf{S}_a$ is any subgroup isomorphic to $\mathbf{S}_{a_1} \times \mathbf{S}_{a_2} \times ... \times \mathbf{S}_{a_k}$, with $(a_1, a_2, ..., a_k)$ a partition of the number a, i.e. with $a_1 + a_2 + ... + a_k = a$.

4 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_1ab_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 partitioned by $\mathbf{H}$, are of different sizes (ranging from h to h^2). The number of double cosets, in which $\mathbf{G}$ is partitioned by $\mathbf{H}$, therefore is not easy to predict. 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 fewer double cosets than there are cosets. This results in small libraries for synthesis.

For synthesizing all members of $\mathbf{S}_8$, Van Rentergem, De Vos and Storme [12] have chosen the double cosets of the above mentioned subgroup chain (1). By its subgroup $\mathbf{S}_4^2$, the group $\mathbf{S}_8$ is partitioned into five double cosets. Van Rentergem et al. conclude that, for synthesizing all 40,320 members of $\mathbf{S}_8$, they need a library of only $(4+2+1)+1=8$ elements. 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 two other double cosets in which $\mathbf{S}_4$ is partitioned by $\mathbf{S}_2\times\ \mathbf{S}_2$. Van Rentergem et al. have demonstrated that it is always possible to construct a representative that is a control gate, i.e. a gate satifying

$$P = f(B, C, ...) \oplus A,$$

together with $Q = B$, $R = C$, ... The control function f is an arbitrary Boolean function. The control gates [13] of width w form a subgroup isomorphic to $\mathbf{S}_2^{2^{w-1}}$ of order $2^{2^{w-1}}$.

Among the control gates, we note three special elements:

- If f is identically zero, then P is always equal to A. Then the gate is the identity gate i.
- If f is identically one, then P always equals $1 \oplus A$. Then the gate is an inverter or `NOT` gate: $P = \texttt{NOT}\ A$.
- If $f(B, C, D, ...)$ equals the $(w-1)$-bit `AND` function $BCD...$, then the gate is called the $\texttt{CONTROLLED}^{w-1}$ `NOT` gate or `TOFFOLI` gate: if and only if $BCD...$ equals 1, then P equals `NOT` A.

We illustrate the Van Rentergem procedure with the example of Table 1b. It is the (non-linear) truth table of the `MILLER` gate, which is considered as a benchmark [9]. Figure 2a gives the result of repeated application of the procedure, until all subcircuits are member of $\mathbf{S}_2$, i.e. are equal to either the 1-bit identity gate or the 1-bit inverter. The nested schematic can easily be translated into a linear chain of control gates: Figure 2b. This circuit

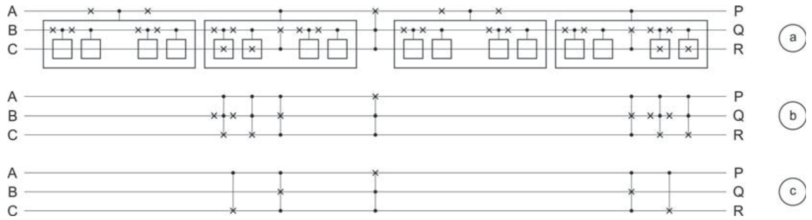

FIGURE 2
Synthesis of the MILLER gate: (a) written as nested conditional gates, (b) written as a linear chain of conditional gates, and (c) after simplification.

consists of four NOT gates and seven CONTROLLED CONTROLLED NOT gates. We now introduce a cost function, called ‘gate cost’: we assign to each control gate a unitary cost. The gate cost of Circuit 2b thus is 11. Figure 2b can be further simplified, yielding Figure 2c of only 5 cost units.

By choosing the subgroup $\mathbf{H}$ isomorphic to $\mathbf{S}_2^4$ (rather than to $\mathbf{S}_4^2$), the supergroup $\mathbf{G} = \mathbf{S}_8$ is partitioned into 282 double cosets. Applying subgroups isomorphic to $\mathbf{S}_2^4$ again and again, leads to a synthesis [14] with a library consisting of all $w \times 2^{2^{w-1}} = 3 \times 16 = 48$ control gates of width $w = 3$. The resulting circuit has length five or less. The synthesis is particularly efficient: for arbitrary width w, the resulting circuit length is only $2w - 1$ or less.

We note that $\mathbf{S}_2^{2^{w-1}}$ and $\mathbf{S}_{2^{w-1}}^2$ are dual Young subgroups of $\mathbf{S}_{2^w}$, as they are based on two dual partitions of the number 2^w :

$$2^w = 2^{w-1} + 2^{w-1}$$
$$2^w = 2 + 2 + ... + 2 \qquad (2^{w-1} \text{ terms}).$$

5 EXPERIMENTAL PROTOTYPES

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 gate realizing the logic gate of Table 1a has six physical inputs: A, NOT A, B, NOT B, C, and NOT C, or, in short-hand notation: A, $\overline{A}$, B, $\overline{B}$, C, and $\overline{C}$. It also has six physical outputs: P, $\overline{P}$, Q, $\overline{Q}$, R, and $\overline{R}$. Such approach is common in electronics, where it is called dual-line or dual-rail electronics.

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 NOTs are NOT gates which are controlled by switches. A first example is the CONTROLLED

NOT gate:

$$P = A$$
$$Q = A \oplus B.$$

These logic relationships are implemented into the 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 3a. In the figure, the arrows show the switch positions if the accompanying label is 1. A second

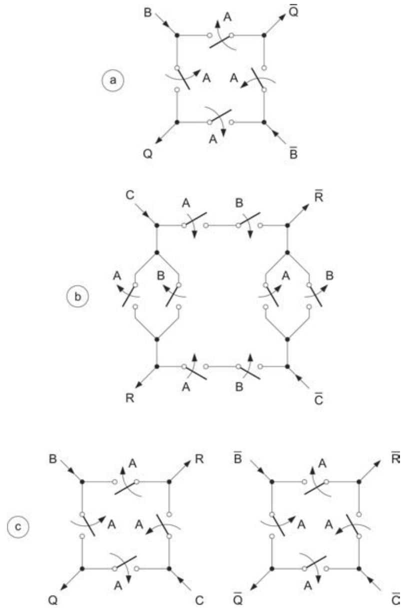

FIGURE 3
Schematic for (a) CONTROLLED NOT gate, (b) CONTROLLED CONTROLLED NOT gate, and (c) CONTROLLED SWAP gate [15].

example is the CONTROLLED CONTROLLED NOT gate or TOFFOLI gate:

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

Its implementation is shown in Figure 3b. The above design philosophy can be extrapolated to a control gate with arbitrary control function f. Suffice it to wire the appropriate series and parallel connection of switches.

Now that we have a hardware approach, we can realize any reversible circuit in hardware. In electronic circuits, a switch is realized by two MOS-transistors in parallel (one n-MOS transistor and one p-MOS transistor). So, for a CONTROLLED NOT, we need 8 transistors and for a CONTROLLED CONTROLLED NOT sixteen.

Switches not only can decide whether an input variable is inverted or not, but equally well decide whether two input variables are swapped or not. This concept leads to the CONTROLLED SWAP or FREDKIN gate:

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

Figure 3c shows the physical realisation, which needs 8 switches, i.e. 16 transistors. A 4-bit ripple adder with twelve CONTROLLED NOTs and four FREDKIN gates contains 160 transistors [15].

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 logic gate [16]. We see how Q becomes smaller and smaller, as transistor dimensions shrink. 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.

Further shrinking of transistor width and length and further reduction of V_t ultimately will lead to a Q value in the neighbourhood 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, they are particularly suited for adiabatic addressing [17], leading to substantial power saving. Figure 4 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 [16].

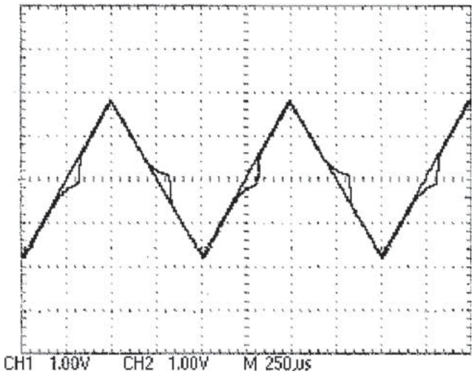

FIGURE 4
Oscilloscope view of 0.35 μm full adder [15].

REFERENCES

[1] Markov, I. (2003). An introduction to reversible circuits. *Proceedings of the Int. Workshop on Logic and Synthesis*, Laguna Beach, 318–319.

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

[3] Hayes, B. (2006). Reverse engineering. *American Scientist 94,* 107–111.

[4] Feynman, R. (1985). Quantum mechanical computers. *Optics News, 11,* 11–20.

[5] Scott, W. (1964). Group theory. Dover Publications (New York).

[6] Patel, K., Markov, I., and Hayes, J. (2004). Optimal synthesis of linear reversible circuits. *Proceedings of the 13 th Int. Workshop on Logic and Synthesis*, Temecula, 470–477.

[7] Sasao, T. (1996). Representations of logic functions using exor operations. In: T. Sasao and M. Fujita : Representations of discrete functions. Kluwer Academic (Boston), 29–54.

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

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

[10] Rentergem, Van Y., De Vos, A., and De Keyser K. (2006). Using group theory in reversible computing. *I.E.E.E. World Congress on Computational Intelligence*, Vancouver, 8566–8573.

[11] Kerber, A. (1970). Representations of permutation groups I. *Lecture Notes in Mathematics*, volume 240, Springer Verlag (Berlin), 17–23.

[12] Van Rentergem, Y., De Vos, A., and Storme, L. (2005). Implementing an arbitrary reversible logic gate. *Journal of Physics A: Mathematical and General, 38,* 3555–3577.

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

[14] De Vos, A., and Van Rentergem Y. (submitted for publication). Young subgroups for reversible computers. *Reports on Mathematical Physics*.

[15] Van Rentergem Y., and De Vos, A. (2005). Optimal design of a reversible full adder. *Int. Journal of Unconventional Computing 1* 339–355.

[16] De Vos, A., and Van Rentergem, Y. (2005). Energy dissipation in reversible logic addressed by a ramp voltage. *Proceedings of the 15 th Int. PATMOS Workshop*, Leuven, 207–216.

[17] Patra, P., and Fussell, D. (1996). On efficient adiabatic design of MOS circuits. *Proceedings of the 4 th Workshop on Physics and Computation*, Boston, 260–269.