Int. Journ. of Unconventional Computing, Vol. 1, pp. 339–355
Reprints available directly from the publisher
Photocopying permitted by license only

Optimal Design of A Reversible Full Adder

YVAN VAN RENTERGEM AND ALEXIS DE VOS*

Universiteit Gent and Imec v.z.w., vakgroep elektronica, Sint Pietersnieuwstraat 41, B - 9000 Gent, Belgium.

Received 9 November 2004; Accepted 17 December 2004

Four designs for reversible full-adder circuits are presented. The implementation of these logic circuits into electronic circuitry is based on c-MOS technology and pass-transistor design. In particular, we investigate the use of the fundamental building block of Fredkin. We present a chip containing three different reversible full adders.

Keywords: reversible computing, Fredkin gate, full adder

1. INTRODUCTION

Classical computing machines using irreversible logic gates unavoidably generate heat. This is due to the fact that each loss of one bit of information is accompanied by an increase of the environment's entropy by an amount $k \log(2)$, where k is Boltzmann's constant. In turn this means that an amount of thermal energy equal to $kT \log(2)$ is transferred to the environment, having a temperature T. According to Landauer's principle [1,2], it is possible to construct a computer that dissipates an arbitrarily small amount of heat [3]. A necessary condition is that no information is thrown away. Therefore, logical reversibility is a necessary (although not sufficient) condition for physical reversibility.

It is widely known that an arbitrary boolean function can be implemented into logic using only NAND-gates. A NAND-gate has two binary inputs (say A and B) but only one binary output (say P), and

* E-mail: alex@elis.UGent.be

therefore is logically irreversible. Fredkin and Toffoli [4] have shown that a basic building block which is logically reversible, should have three binary inputs (say A, B, and C) and three binary outputs (say P, Q, and R).

Feynman [5,6] has proposed the use of three fundamental gates:

- the NOT gate,
- the CONTROLLED NOT gate, and
- the CONTROLLED CONTROLLED NOT gate.

Together they form a set of three building blocks with which we can synthesize an arbitrary logic function. The NOT gate simply realizes

$$P = \text{NOT } A.$$

The CONTROLLED NOT satisfies

$$P = A$$

together with

$$\begin{aligned} &\text{If } A = 0, \quad \text{then} \quad Q = B, \\ &\qquad\qquad\ \ \text{else} \quad Q = \text{NOT } B. \end{aligned} \tag{1}$$

See Table 1a. The CONTROLLED CONTROLLED NOT gate (a.k.a. the TOFFOLI gate) satisfies

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

TABLE 1
Truth tables: (a) CONTROLLED NOT, (b) CONTROLLED CONTROLLED NOT, (c) FREDKIN.

A	B	P	Q
0	0	0	0
0	1	0	1
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	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

(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	1	0
1	1	0	1	0	1
1	1	1	1	1	1

(c)

together with

$$\begin{aligned} &\text{If } A \text{ AND } B = 0, \quad \text{then} \quad R = C, \\ &\qquad\qquad\qquad\quad \text{else} \quad R = \text{NOT } C. \end{aligned} \tag{2}$$

See Table 1b.

The logic expressions of the CONTROLLED NOT are equivalent with

$$\begin{aligned} P &= A \\ Q &= A \text{ XOR } B, \end{aligned}$$

where XOR is the abbreviation of the EXCLUSIVE OR function. The logic expressions of the CONTROLLED CONTROLLED NOT are equivalent with

$$\begin{aligned} P &= A \\ Q &= B \\ R &= (A \text{ AND } B) \text{ XOR } C. \end{aligned}$$

The CONTROLLED CONTROLLED NOT has a particular property: it 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.

In spite of the fact that the CONTROLLED CONTROLLED NOT is sufficient for building any boolean function, often the three Feynman blocks are used together for synthesis. The NOT block is trivial, as we make use of dual electronics. This means that any boolean variable A is represented by two electric signals: one representing A, the other representing $\overline{A}$ = NOT A. Then, a simple metal cross-over is sufficient to realize the NOT function: P being connected to $\overline{A}$ and $\overline{P}$ to A. Function (1) leads to the implementation of Figure 1a, whereas function (2) is realized as in Figure 1b. The latter circuit is deduced from the former. In the four sides of the square of Figure 1b, the single switches from Figure 1a are replaced either by a series or by a parallel connection.

Storme et al. [7,8] have shown that, among the 40,320 different reversible gates with three inputs and three outputs, there are 38,976 gates that can be used as universal building block. Besides the CONTROLLED CONTROLLED NOT of Feynman, also the FREDKIN gate from Fredkin and Toffoli [4] is a candidate. Its truth table is shown in Table 1c. The outputs satisfy

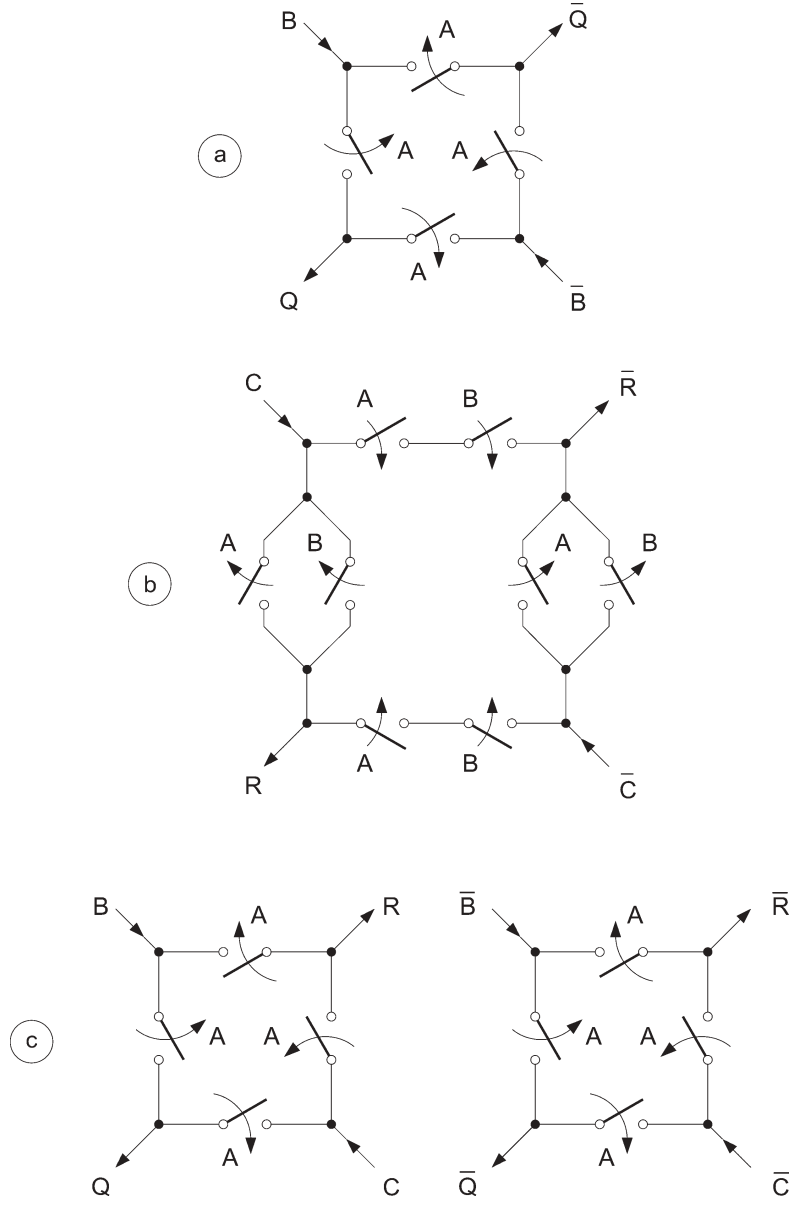


FIGURE 1
Basic square circuits: (a) $Q = A$ XOR B, (b) $R = (A$ AND $B)$ XOR C, (c) $Q = ((\text{NOT } A)$ AND $B)$ OR $(A$ AND $C)$ with $R = ((\text{NOT } A)$ AND $C)$ OR $(A$ AND $B)$.
Note: here a switch is in the state indicated by the arrow if the logic variable next to it equals one.

$P = A$
$Q = ((\text{NOT } A) \text{ AND } B) \text{ OR } (A \text{ AND } C)$
$R = ((\text{NOT } A) \text{ AND } C) \text{ OR } (A \text{ AND } B),$

which can also be interpreted as

$$P = A$$

together with

$$\begin{array}{lll} \text{If } A = 0, & \text{then} & Q = B \\ & & R = C, \\ & \text{else} & Q = C \\ & & R = B. \end{array} \tag{3}$$

This leads to the implementation of Figure 1c.

2. ELECTRONICS

Within the framework of the European multiproject-wafer service *Europractice*, silicon prototypes of some circuits have been fabricated, in the *AMI Semiconductor* n-well c-MOS 0.35 μm technology. The layout is designed with Cadence DesignFrameWork II 4.4.6 full-custom software. The n-MOS transistors have length L equal to 0.35 μm and width W equal to 0.5 μm, whereas the p-MOS transistors have $L = 0.35$ μm and $W = 1.5$ μm. The p-MOS transistors are chosen three times as wide as the n-MOS transistors in order to compensate for the fact that holes are about three times less mobile in silicon than electrons ($\mu_p \approx 500$ cm^2/Vs, whereas $\mu_n \approx 1500$ cm^2/Vs). The threshold voltage V_t is 0.6 V for the n-MOS and –0.6 V for the p-MOS transistors.

For the implementation of the on-off switch, we use a small c-MOS circuit called the transmission gate: see Figure 2c, where V_c denotes the control voltage of the switch. A transmission gate consists of two MOS transistors in parallel, i.e. one n-MOS transistor and one p-MOS transistor. This leads to the following number of transistors:

- the NOT gate: no transistors
- the CONTROLLED NOT gate: 8 transistors
- the CONTROLLED CONTROLLED NOT gate: 16 transistors
- the FREDKIN gate: 16 transistors.

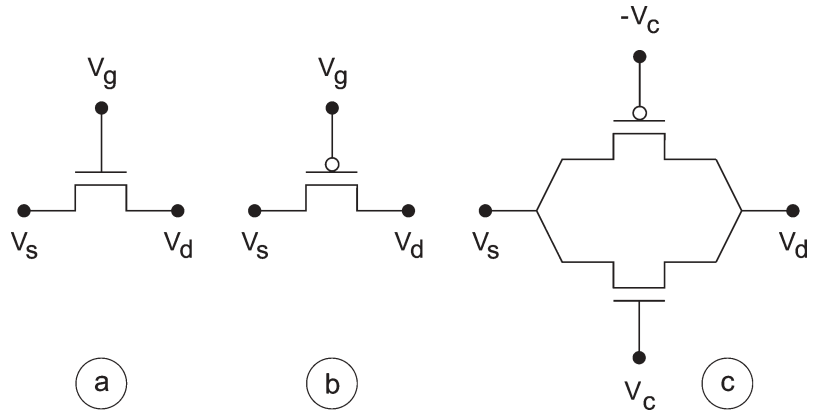


FIGURE 2
Schematic icons: (a) n-MOS transistor, (b) p-MOS transistor, (c) c-MOS transmission gate.

Figure 3 shows the detailed electronic schematic of the four building-blocks. Figure 4 shows the layout of the prototype FREDKIN gate (23 μm $\times$ 14 μm). We stress that these circuits have no power supply inputs. Thus there are neither V_{dd} nor V_{ss} nor ground busbars. Note also the complete absence of clock lines. Thus all signals (voltages and currents) and all energy provided at the outputs originate from the inputs. The inputs receive their power from

- either ideal (time-dependent) voltage sources, or
- an input memory register, built from LC resonant circuits [9–11], or
- an interface with standard c-MOS circuits [12].

The logic circuits are examples of dual-line pass-transistor logic, as opposed to conventional so-called restoring logic. In conventional c-MOS circuits, output pins are fuelled from a V_{dd} and a V_{ss} power line.

The absence of clocks does not mean that our reversible MOS (or r-MOS) circuits belong to the asynchronous logic family. Indeed, neither Rendezvous circuits (be it Muller C-elements or GasP modules) nor Arbiter circuits [13] control the information flow. Although the r-MOS networks are digital, the propagation of information (and energy) mostly resembles the flow of results through an analog computer. A linear chain of transmission gates is a good model [14].

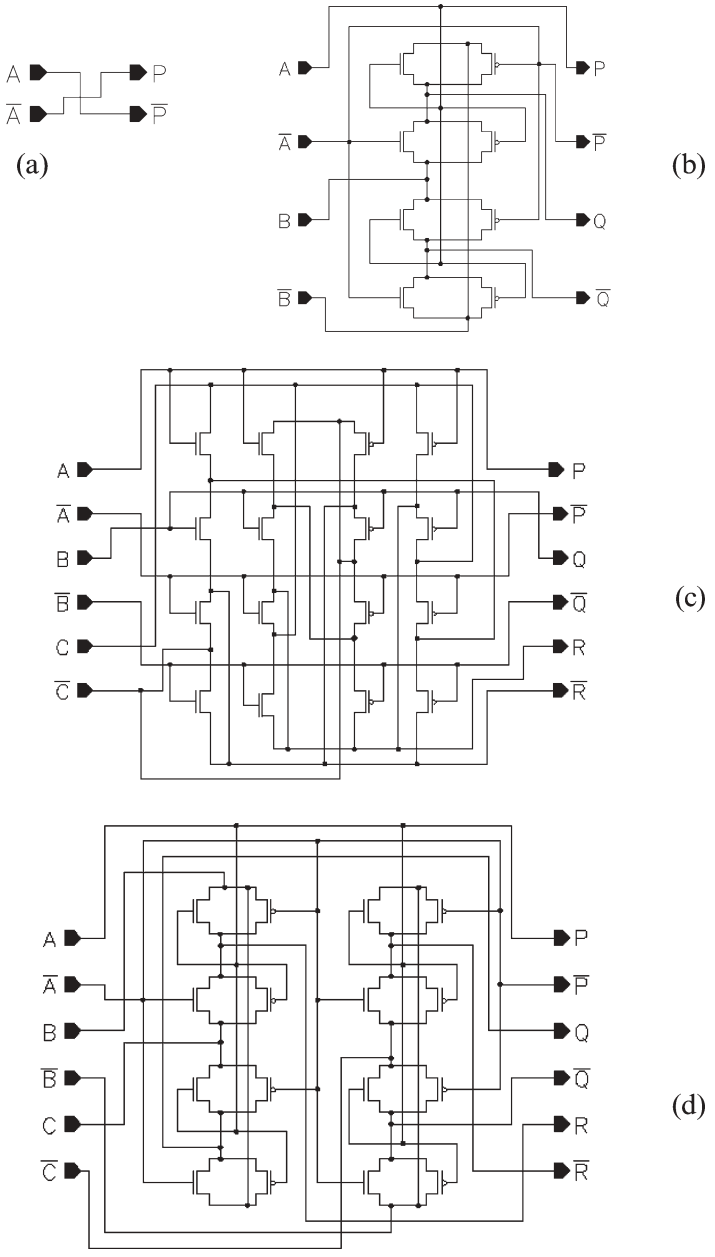


FIGURE 3
Cadence schematic of basic circuits: (a) the NOT gate, (b) the CONTROLLED NOT gate, (c) the CONTROLLED CONTROLLED NOT gate, and (d) the FREDKIN gate.

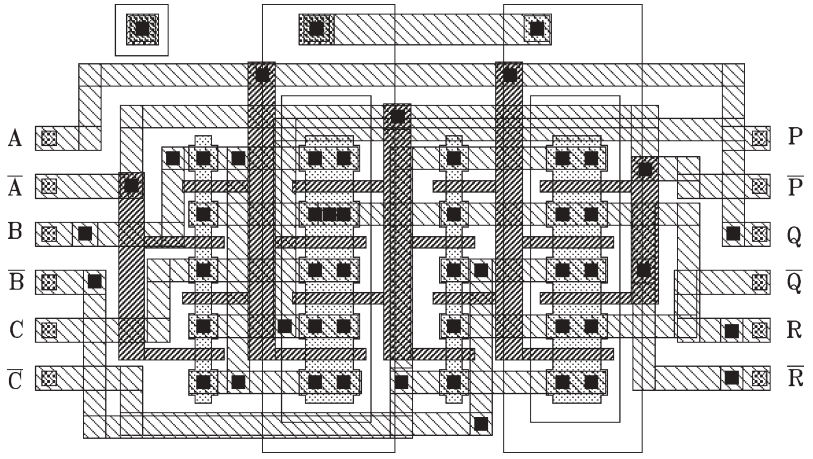


FIGURE 4
The FREDKIN gate.

TABLE 2
The full adder: (a) conventional truth table, (b) reversible truth table (using one preset P and two garbages G_1 and G_2).

A	B	C_i	C_o	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(a)

A	B	C_i	P	C_o	S	G_1	G_2
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	1	1	0	1
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1
1	0	0	1	1	1	1	1
1	0	1	0	1	0	1	1
1	0	1	1	0	0	1	1
1	1	0	0	1	0	1	0
1	1	0	1	0	0	1	0
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0

(b)

3. APPLICATION

Higher levels of computation particularly need the implementation of the full adder. Table 2a shows the truth table, where the input C_i denotes

the carry-in bit. The table says that $0 + 0 + 0 = 0$, $0 + 0 + 1 = 1, \ldots, 1 + 1 + 0 = 2$, and $1 + 1 + 1 = 3$.

Desoete et al. [15] have presented a silicon implementation of the full adder, applying two CONTROLLED NOTs and two CONTROLLED CONTROLLED NOTs. See Figure 5a. The circuit thus contains $2 \times 8 + 2 \times 16 = 48$ transistors.

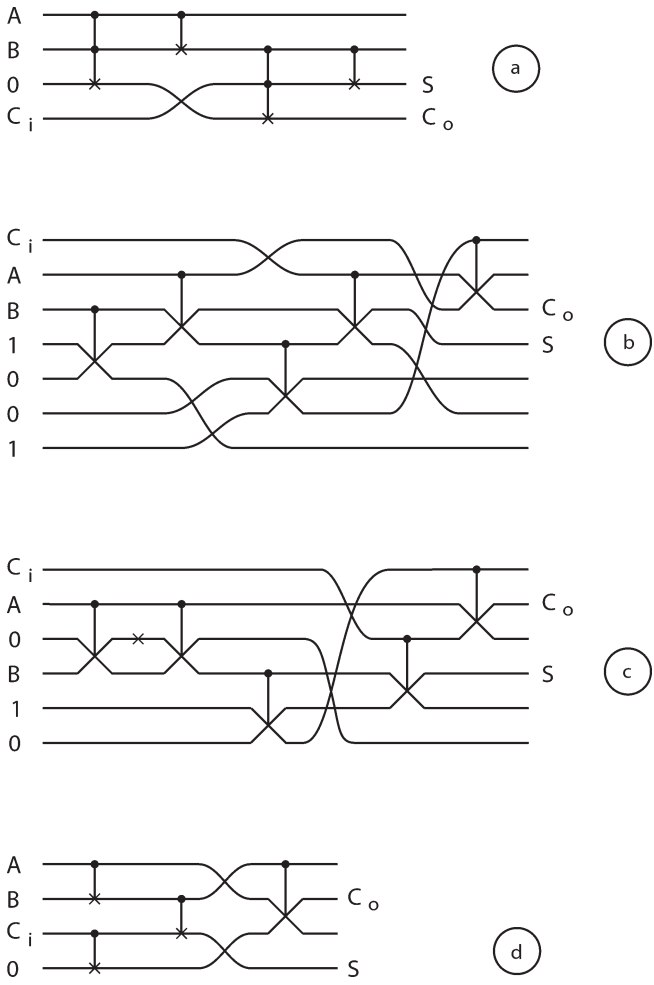


FIGURE 5
Four different full adders : (a) with two CONTROLLED NOTs and two CONTROLLED CONTROLLED NOTs, (b) with five FREDKINs, (c) with five FREDKINs and one NOT, (d) with three CONTROLLED NOTs and one FREDKIN.

Besides the inputs A, B, and C_i, there is a fourth, constant input: the so-called preset. Besides the sum bit S and the carry-out bit C_o, the full adder also provides a third and a fourth output bit. These results are considered as 'garbage'. The garbage outputs are the counterpart of the preset inputs. Presets and garbages are necessary in order to guarantee logic reversibility. Indeed, Table 2a is not reversible. If we forget the value of the input (A, B, C_i), knowledge of the output (C_o, S) is not sufficient to recover the input. E.g. (C_o, S) = (0, 1) can correspond to the input (A, B, C_i) = (0, 0, 1), but can equally well be the result of (A, B, C_i) = (0, 1, 0) or (A, B, C_i) = (1, 0, 0). We have to add at least two extra output columns, in order to realise a one-to-one mapping between inputs and outputs. The existence of four output bits then automatically necessitates a fourth input bit (i.e. the preset). There is a lot of freedom for the introduction of the fourth input column and the third and fourth output columns. Table 2b gives an example: the reversible truth table used in Figure 5a. The reader will easily verify that Table 2b fulfils two conditions: (a) all sixteen outputs ($C_oSG_1G_2$) are different, such that the table is reversible, and (b) if $P = 0$, then the output colums C_o and S have the values as in Table 2a. The outputs ($C_oSG_1G_2$) contain the same amount of information as the inputs (ABC_iP), such that backward calculation is possible. If desired, it is always possible to copy and store the required outputs (C_oS) and perform the reverse calculation, such that the garbage bits (G_1G_2) are recycled [4]. However, in our prototype chip, we have not implemented such 'spy' and 'undo' circuits.

Bruce et al. [16] have proposed four different implementations of the full adder, all based on the FREDKIN gate. Two designs need five FREDKIN gates, the other two need only four FREDKINs. However, three of these circuits cannot be considered as truly reversible. Indeed, two designs (their Figures 5 and 8) contain one or more fan-outs, whereas a third one (their Figure 6) contains a loop. Neither fan-outs nor loops are allowed in reversible combinatorial circuits. Only Figure 9 of Bruce et al. is a truly reversible FREDKIN full adder: see our Figure 5b. It uses five Fredkin gates.

We propose here an alternative design, using an equal number of Fredkin gates: see Figure 5c. Here, the number of binary inputs (equal to the number of binary outputs) is only six, in contrast to the Bruce design, which needs seven inputs and seven outputs. Less inputs and less outputs mean less interconnections to the outside world. The price, we have to pay for this advantage, is an additional NOT gate. As, in our technology, NOT gates are for free (i.e. need zero transistors), this disadvantage is

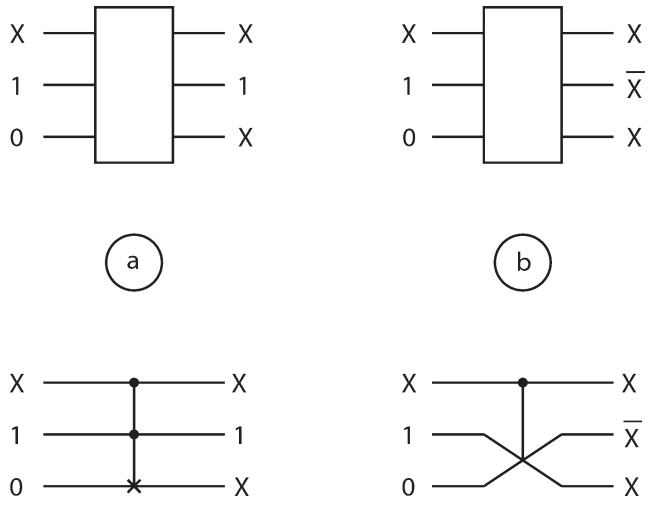


FIGURE 6
Duplicating: (a) with CONTROLLED CONTROLLED NOT, (b) with FREDKIN.

much smaller than the cost reduction in wiring, bonding and packaging [17].

Both Figures 5b and 5c contain five FREDKIN gates and thus need 80 transistors. Comparison with the 48 transistors of Figure 5a makes clear that the FREDKIN gate is not very efficient for implementing a full adder. One of the reasons is that reversible circuits are not allowed to make use of fan-out and therefore need reversible 'duplicators' [18]. Figure 6 shows how to duplicate an arbitrary boolean variable X with the help of either a CONTROLLED CONTROLLED NOT or a FREDKIN gate and two constant inputs (i.e. two presets). The CONTROLLED CONTROLLED NOT just duplicates X, the third output being a constant, which can be 'recycled', i.e. can be used as an input further downstream. The FREDKIN gate, however, needlessly triplicates X, yielding a third (useless) copy of X.

Reversible full-adder design becomes much more compact if we combine FREDKIN gates with CONTROLLED NOT gates. Figure 5d shows a particularly efficient design: a full adder needing only three CONTROLLED NOTs and a single FREDKIN, thus using only 40 transistors. Brute-force exhaustive search with the help of the computer algebra package Maple verified that no reversible design with fewer transistors is possible. For that reason, we call this design 'optimal'.

TABLE 3
The full adder.

design	width	depth	# gates	# transistors
a	4	2	4	48
b	7	5	5	80
c	6	5	6	80
d	4	2	4	40

Table 3 compares the four designs of Figure 5. Besides the number of logic gates and the number of transistors, the table also gives the width of the circuit (i.e. the number of logic inputs, which is also the number of logic outputs) and the logic depth of the circuit. The last property is the number of subsequent elementary computational steps, necessary to obtain the final result. Its precise definition is given in reference [17]. All four cost numbers should be as small as possible. It is clear from the table that design d outperforms the three others.

Figure 7 shows the results of Spectre simulations for the four full adders of Figure 5, in case of an adiabatic addressing of the circuits

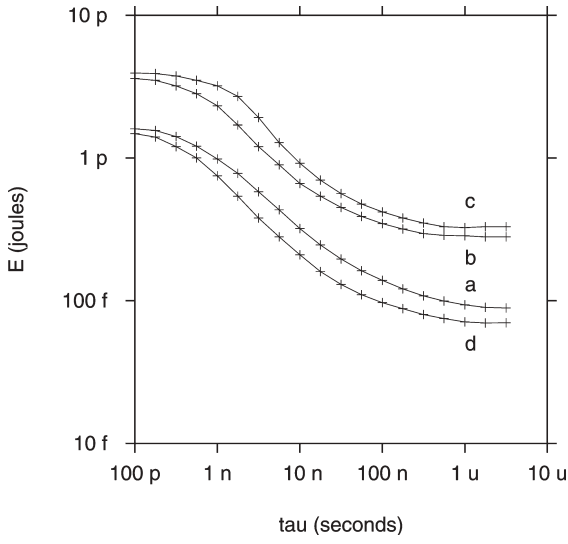


FIGURE 7
Spectre simulation of the four full adders: (a) Feynman adder, (b) Fredkin adder, (c) Fredkin adder, and (d) mixed adder.

[9,19–21]. The energy dissipation E per computational cycle is calculated as a function of the addressing time τ. In this cycle, the input (A, B, C_i) switches adiabatically from (1, 0, 0) to (1, 1, 0) and back. We see how the Fredkin adders consume 2.5 to 3.0 times more energy than the Feynman adder. The 'mixed' design needs only 0.8 times the energy needed by the Feynman design. These four energy consumptions are roughly in proportion to the respective transistor counts. Figure 8 shows the layout of three different full adders (about 0.30 mm × 0.17 mm). The complete chip (i.e. bonding pads and protection circuits included) measures 1.5 mm × 0.9 mm. Finally, Figure 9 displays the experimental transient signals (for τ equal 500 μs) of input bit B and carry-out bit C_o. Input power is delivered by a standard function generator. We clearly see that the output faithfully follows the input, except in the voltage range $\pm V_t/2$. During the positive ramp, energy flows from the function generator to the output capacitor; during the negative ramp, most (i.e. about 95%) of the energy flows back from the output capacitor to function generator.

For sake of comparison with Figure 9d, Figure 10 gives the output of a circuit simulation with Spectre. The good agreement between theory and experiment illustrates the reliability of the simulations.

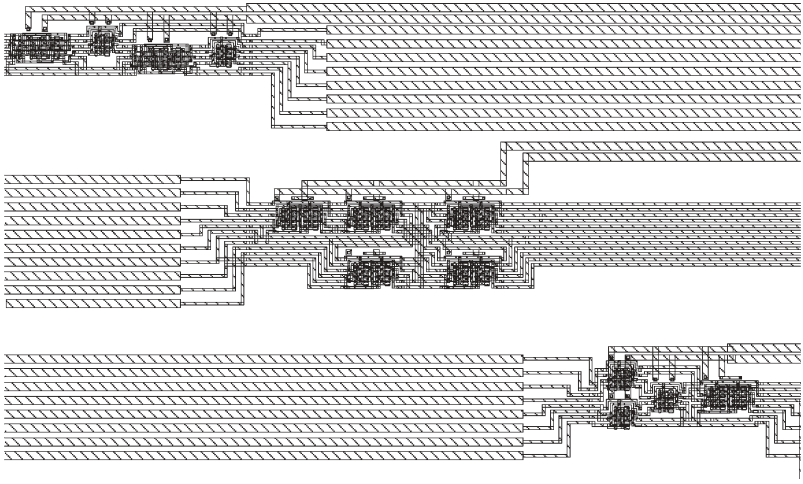
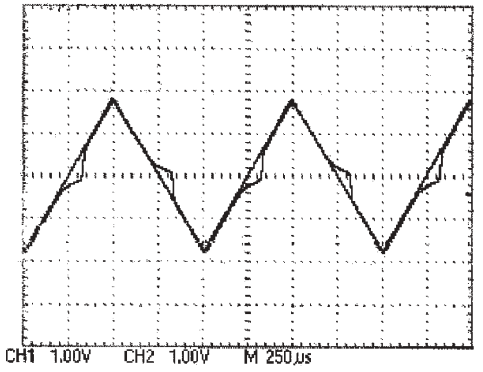
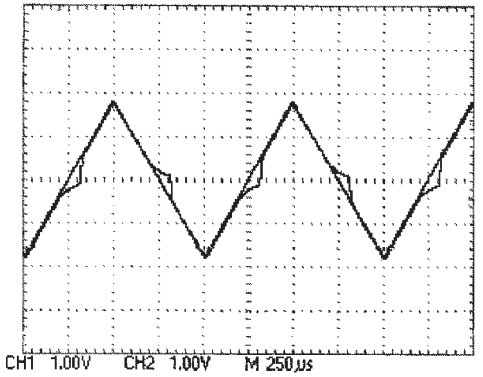


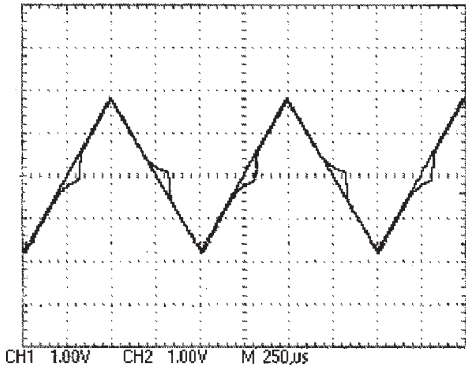
FIGURE 8
Three different full adders (designs a, c, and d) in 0.35 μm c-MOS technology.



(a)



(c)



(d)

FIGURE 9
Oscilloscope view of three full adders: (a) Feynman adder, (c) Fredkin adder, and (d) mixed adder.

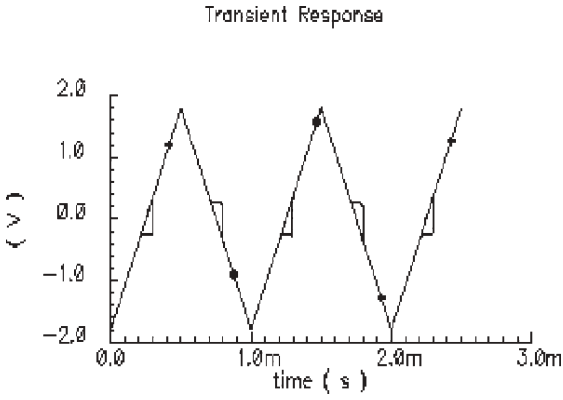


FIGURE 10
Spectre simulation of optimal (i.e. mixed) full adder.

4. CONCLUSION

We have designed four different reversible full adders. The designs imply that their outputs contain as much information as their inputs and thus, according to Landauer's principle, that they (at least in principle) can operate with arbitrarily low dissipation. We have optimized the circuit with the help of the fundamental building block proposed by Fredkin. The electronic implementation of this logic gate is based on dual-line pass-transistor logic. An important characteristic of the circuit is that all energy supplied to the system is delivered by the input signals themselves. Three of the four full adders have been implemented into silicon. The optimal design contains three CONTROLLED NOT gates and one FREDKIN gate, i.e. only forty transistors, and has a logic width of four and a logic depth of two. Its surface area is about 6 μm $\times$ 3 μm in a standard 0.35 μm c-MOS technology.

The full adder should be regarded as only a benchmark. Indeed, both the CONTROLLED CONTROLLED NOT and the FREDKIN are universal primitives. This means that any reversible combinatorial network can be built using a finite number of one of these two building blocks. We have shown however that it is advantageous to apply also the NOT and the CONTROLLED NOT. Therefore, we conclude that a library consisting of four standard cells (NOT, CONTROLLED NOT, CONTROLLED CONTROLLED NOT, and FREDKIN) is

ideal for designing any reversible VLSI computer. When, in future deep-submicron technologies, V_t will continue to shrink, reversible architecture and adiabatic addressing will become profitable to VLSI computers.

Acknowledgement

The authors thank the *Invomec* division of *Imec v.z.w.* (Leuven, Belgium) and the *Europractice* organisation, for processing the chips at *AMI Semiconductor* (Oudenaarde, Belgium).

REFERENCES

[1] Bennett, C. and Landauer, R. (July 1985). The fundamental physical limits of computation. *Scientific American, 253*, 38–46.

[2] Stix, G. (September 1998). Riding the back of electrons. *Scientific American, 279*, 20–21.

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

[4] Fredkin, E. and Toffoli, T. (1982). Conservative logic. *Int. Journal of Theoretical Physics, 21*, 219–253.

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

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

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

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

[9] Athas, W., Tzartzanis, N., Svensson, L. and Peterson, L. (1997). A low-power microprocessor based on resonant energy. *IEEE Journal of Solid-State Circuits, 32*, 1693–1701.

[10] Lo, C. and Chan, P. (1999). An adiabatic differential logic for low-power digital sytems. *IEEE Transactions on Circuits and Systems — II. Analog and Digital Signal Processing, 46*, 1245–1250.

[11] Devisch, F., Stiens, J., Vounckx, R. and Kuijk, M. (2000). Low power RLC-resonant method for driving liquid-crystal displays. *Digest of Technical Papers of the SID Int. Symposium.* Long Beach, 594–597.

[12] Amirante, E., Fischer, J., Lang, M., Bargagli–Stoffi, A., Berthold, J., Heer, C. and Schmitt–Landsiedel, D. (2003). An ultra low-power adiabatic adder embedded in a standard 0.13 μm CMOS environment. *Proc. 29th Eur. Solid-State Circuits Conf.* Estoril, 599–602.

[13] Sutherland, I. and Ebergen, J. (August 2002). Computers without clocks. *Scientific American, 287*, 46–53.

[14] Alioto, M. and Palumbo, G. (2002). Analysis and comparison on full adder block in submicron technology. *IEEE Transactions on Very Large Scale Integration Systems.* 806–823.

[15] Desoete, B., De Vos, A., Sibiński, M. and Widerski, T. (1999). Feynman's reversible logic gates, implemented in silicon. *Proc. 6th Int. Conf. Mixdes.* Kraków, 497–502.

[16] Bruce, J., Thornton, M., Shivakumaraiah, L., Kokate, P. and Li, X. (2002). Efficient adder circuits based on a conservative reversible logic gate. *Proc. IEEE Computer Society Annual Symposium on VLSI.* Pittsburgh, 83–88.

[17] De Vos, A. and Beunis, F. (2005). Designing reversible logic. To be published.

[18] Kerntopf, P. (2002). On universality of binary reversible logic gates. *Proc. 5th Workshop on Boolean Problems.* Freiberg, 47–52.

[19] Athas, W., Svensson, L., Koller, J., Tzartzanis, N. and Chou, E. (1994). Low-power digital systems based on adiabatic-switching principles. *IEEE Transactions on Very Large Scale Integration Systems, 2*, 398–407.

[20] Ye, Y. and Roy, K. (1996). Energy recovery circuits using reversible and partially reversible logic. *IEEE Transactions on Circuits and Systems — I. Fundamental Theory and Applications, 43*, 769–778.

[21] De Vos, A. and Desoete, B. (2000). Equipartition principles in finite-time thermodynamics. *Journal of Non-Equilibrium Thermodynamics, 25*, 1–13.