

On Digital LSI Circuits Exploiting Collision-Based Fusion Gates

KAZUHITO YAMADA, TETSUYA ASAI*, TETSUYA HIROSE,
AND YOSHIHITO AMEMIYA

*Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814 Japan
E-mail: asai@sapiens-ei.eng.hokudai.ac.jp*

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

A novel circuit-design method for low-power and compact digital LSI circuits based on collision-based reaction-diffusion computing is proposed. We show that i) fundamental logic gates can be constructed by a small number of our unit gates, ii) multiple-input logic gates are constructed in a systematic manner and iii) the number of transistors in specific logic gates constructed by the proposed method is significantly smaller than that of conventional logic gates while maintaining low-power operations.

Keywords: digital circuits, integrated circuits, low-power circuits, collision-based computing, reaction-diffusion systems

1 INTRODUCTION

Recently, power consumption of high performance digital microprocessors is rapidly increasing. Indeed, look at a trend line borrowed from Intel indicates the future power density of microprocessors exceeding the power density on the surface of the sun. Demands for low-power microprocessors are thus growing at a high rate. The easiest way to achieve low-power consumption in such digital LSI circuits is to decrease the power-supply voltage because the power consumption is proportional to the square of the supply voltage. However, combinational logic circuits implemented in digital LSIs exploit a number of metal-oxide semiconductor field-effect transistors (MOS FETs) where many MOS FETs are on the current path between the power supply and the ground. Therefore, due to stacking

effects of the MOS FETs' nonzero threshold voltages, the supply voltage cannot easily be decreased, even though the threshold voltage is decreasing as LSI fabrication technology advances year by year.

Any logic function can be constructed by combining multiple two-input NAND gates. The NAND gate consists of four MOS FETs where three MOS FETs are on the current path between the power supply and the ground. However, practical logic gates for integrated circuits use the simplest two-input NAND gates'@very rarely, but use special logic gates frequently, each of which has more than three MOS FETs on the current path, to minimize the circuit area on a chip. Therefore, decreasing such circuits' supply voltage is difficult compared with two-input NAND-based logic circuits. But, if only NAND gates are used to decrease the supply voltage, a large number of the gates will be required for constructing complex logic functions.

Several designing methods have been proposed for implementing complex logic functions with small number of MOS FETs. The Reed-Muller expansion [12, 11], which expands logical functions into combinations of AND and XOR logic, enables us to design 'specific' arithmetic functions with a small number of gates, but it is not suitable for arbitrary arithmetic computation. Pass-transistor logic (PTL) circuits use a small number of MOS FETs for basic logic functions but additional level-restoring circuits are required for every unit [16]. Current-mode logic circuits also use a small number of MOS FETs for basic logic, but their power consumption is very high due to the continuous current flow in turn-on states [3]. Subthreshold logic circuits where all the MOS FETs operate under their threshold voltage are expected to exhibit ultra-low power consumption, but the operation speed is extremely slow [14, 15]. Binary decision diagram logic circuits are suitable for next-generation semiconductor devices such as single-electron transistors [13, 18, 4], but not for present digital LSIs because of the use of PTL circuits.

To address the problems above concerning low-power operation with practical operation speed in digital LSI circuits, we describe a method of designing logic circuits with collision-based fusion gates, which is inspired by collision-based reaction-diffusion computing (RDC) [1, 10, 9]. This paper is organized as follows. In section 2, we introduce a new interpretation of collision-based RDC, especially concerning directions and speeds of propagating information quanta. Then in section 3, we exhibit basic logic functions constructed by simple unit operators, i.e., fusion gates, and compare the number of MOS FETs used in low-power conventional logic circuits and our fusion-gate logic circuits. Possible usable architectures for a digital multiplier and a 4-bit microprocessor constructed by our fusion gates are presented in Section 4. Section 5 is a summary.

2 NEW INTERPRETATION OF COLLISION-BASED COMPUTING FOR DIGITAL LSIS

Dynamic, or collision-based, computers employ mobile self-localizations, which travel in space and execute computation when they collide with each other. Truth values of logical variables are represented by the absence or presence of the traveling information quanta. There are three sources of collision-based computing: proof of the universality of Conway's Game of Life via collisions of glider streams [5], conservative logic [6], cellular automaton implementation of the billiard ball model [7], and particle machine [17] (a concept of computation in cellular automata with soliton-like patterns); see overviews in [1].

The main purpose of collision-based computing is to perform computation in an 'empty space', i.e., a medium without geometrical constraints. Basic toy models of collision-based computing are shown in Fig. 1. In the billiard ball logic shown in Fig. 1(a), a set of billiard balls are fired into a set of immovable reflectors at a fixed speed. As the billiard balls bounce off each other and off the reflectors, they perform a reversible computation. Provided that the collisions between the billiard balls and between the billiard balls and the reflectors are perfectly elastic, the computation can proceed at a fixed finite speed with no energy loss.

Adamatzky demonstrated that a similar computation can be performed on excitable reaction diffusion systems [1, 2]. Figure 1(b) illustrates basic logic gates where instead of billiard ball wave fragments (white localizations in the figure) travel in an excitable reaction-diffusion medium. In typical excitable media, localized wave fragments facing each other disappear when they collide. With a special setup described in [2], those excitable waves do not disappear, but they do produce subsequent excitable waves.

Our basic idea here is considering "What happens if wave fragments travel in limited directions instantaneously?" For example, when such wave fragments are generated at the top and end of a pipe (not an empty space)

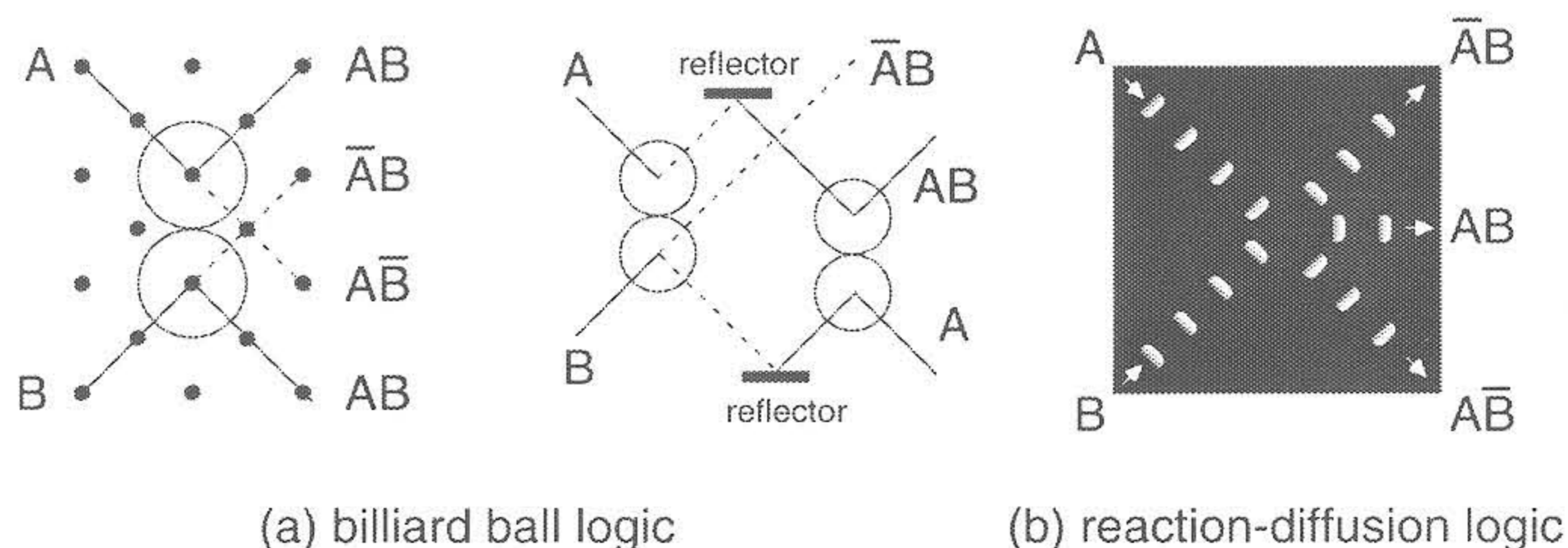


FIGURE 1

Collision-based computing models (a) conservative billiard-ball logic for AND and partial XOR computation [6, 7] and (b) nonconservative (dissipative) reaction-diffusion logic that has the same function as that of (a) [2].

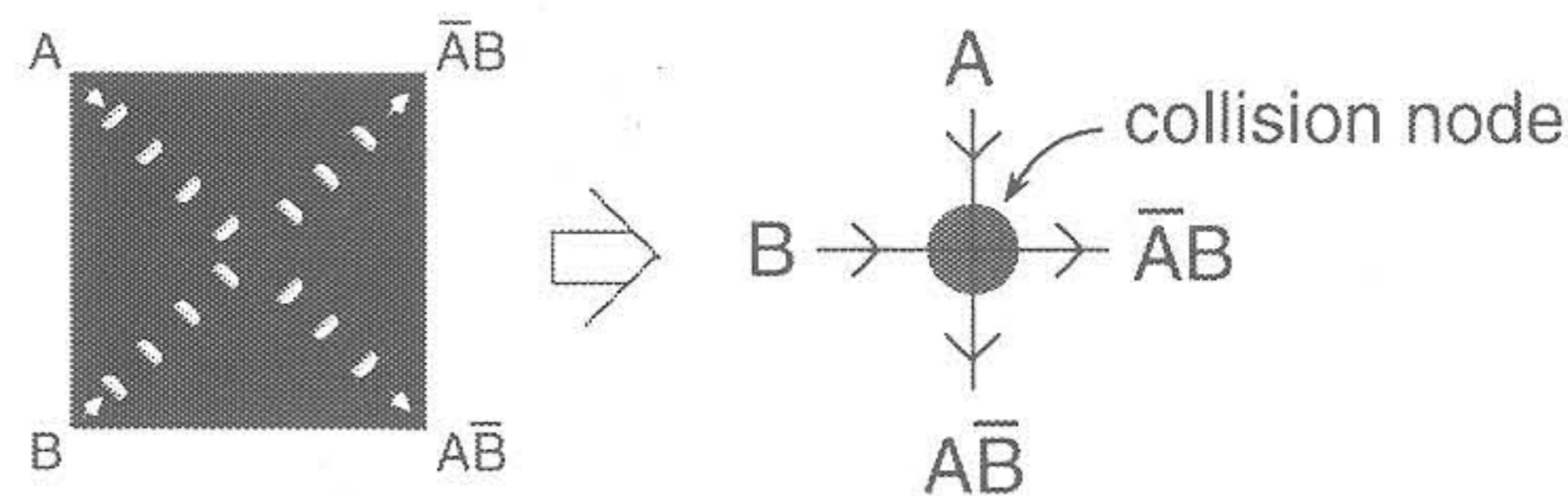


FIGURE 2
Definition of collision-based fusion gate.

filled with excitable chemicals, these waves may disappear at the center of the pipe instantaneously. When two pipes are perpendicularly arranged and connected, wave fragments generated at the tops of the two pipes may also disappear at the connected point. If only one wave fragment (A or B) is generated at the top of one pipe, it can reach the end of the pipe [\overline{AB} or $A\overline{B}$ in Fig. 1(b)]. A schematic model of this operation is shown in Fig. 2. In Fig. 2 left, an excitable reaction-diffusion medium, where excitable waves (A and B) may disappear when they collide, is illustrated. In Fig. 2 right, an equivalent model of two perpendicular directions of wave fragments, i.e., North-South and West-East fragments, is depicted. The input fragments are represented by values A and B where A (or B) = "1" represents the existence of a wave fragment traveling North-South (or West-East), and A (or B) = "0" represents the absence of wave fragments. When $A = B = "1"$ wave fragments collide at the center position (black circle) and then disappear. Thus, East and South outputs are "0" because of the disappearance. If $A = B = "0"$, the outputs will be "0" as well because of the absence of the fragments. When $A = "1"$ and $B = "0"$, a wave fragment can travel to the South because it does not collide with a fragment traveling West-East. The East and South outputs are thus "0" and "1", respectively, whereas they are "1" and "0", respectively, when $A = "0"$ and $B = "1"$. Consequently, logical functions of this simple 'operator' are represented by \overline{AB} and $A\overline{B}$, as shown in Fig. 2 right. We call this operator a 'collision-based fusion gate', where two inputs correspond to perpendicular wave fragments, and two outputs represent the results of collisions (transparent or disappear) along the perpendicular axes. Notice that in this configuration the computation is performed with geometrical constraints.

3 DIGITAL CIRCUITS WITH COLLISION-BASED FUSION GATES

A collision-based fusion gate receives two logical inputs (A and B) and produces two logical outputs; i.e., \overline{AB} and $A\overline{B}$. The corresponding MOS circuit receives logical (voltage) inputs (A and B) and produce these logic functions, as shown in Fig. 3(a) left. The minimum circuit is designed based on PTL circuits where a single-transistor AND logic is fully utilized. In

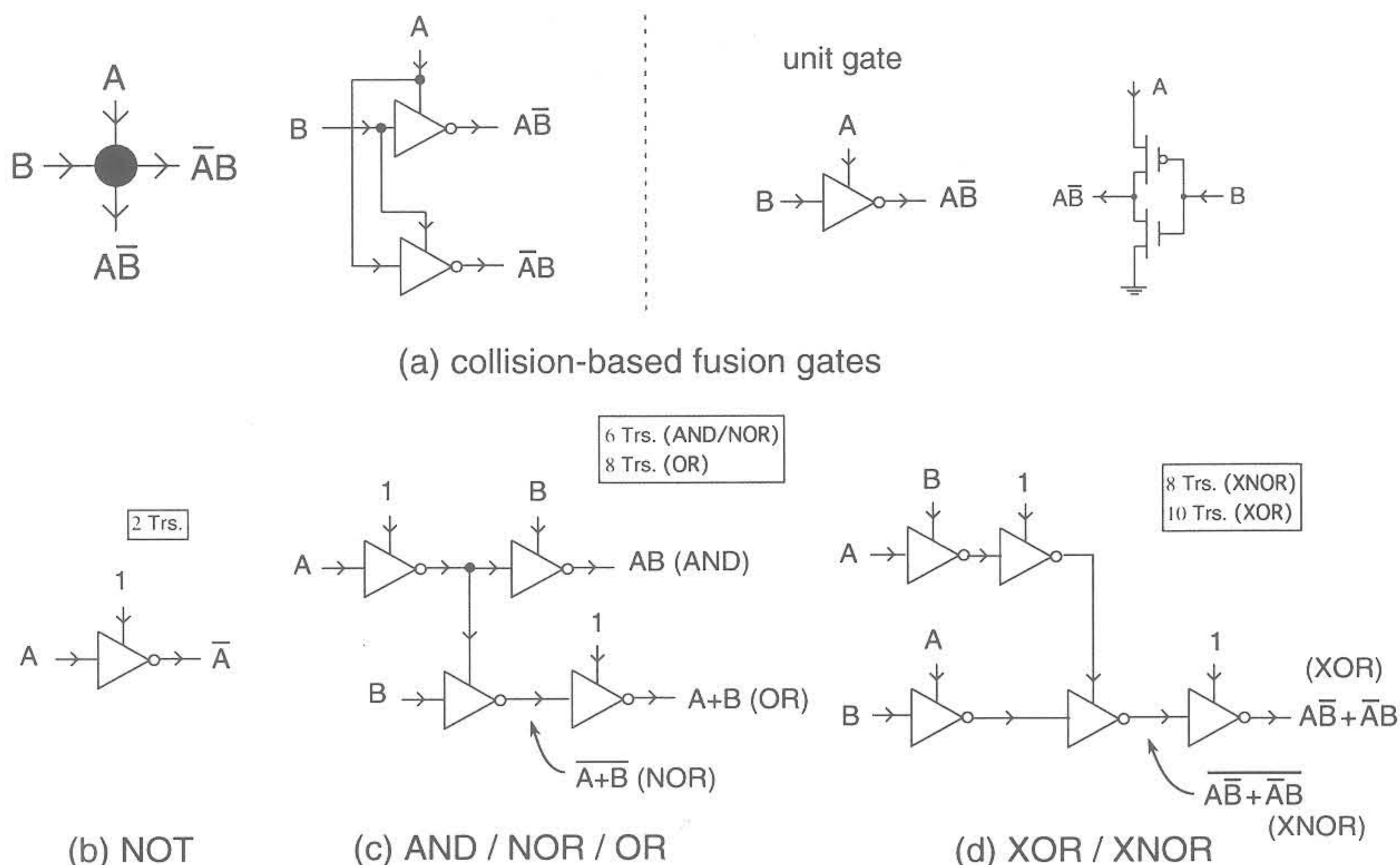


FIGURE 3 MOS circuits for collision-based fusion gate (a) and basic logical circuits using several units [(b)-(d)] that produce multiple logical functions.

Fig. 3(a) right, a pMOS pass transistor is responsible for the $\bar{A}B$ function, and an additional nMOS FET is used for discharging operations. When the pMOS FET receives voltages A and B at its gate and drain, respectively, the source voltage approaches $\bar{A}B$ at equilibrium. If a pMOS FET is turned off, an nMOS FET connected between the pMOS FET and the ground discharges the output node, which significantly increases the upper bound of the operation frequency.

Figures 3(b) to (d) represent basic logic circuits constructed by combining several fusion gates. The simplest example is shown in Fig. 3(b) where the NOT function is implemented by a fusion gate. The North input is always “1”, whereas the West is the input (A) of the NOT function. The output appears on South node (\bar{A}). Figure 3(c) represents a combinational circuit of fusion gates that produces AND, NOR and OR functions. An OR function is obtained by combining NOT and AND/NOR fusion gates. Exclusive logic functions are produced by four (for XNOR) or five (for XOR) fusion gates as shown in Fig. 3(d).

Figure 4 shows constructions of multiple-input logic functions with our fusion gates. In classical circuits, two-input AND and OR gates have six MOS FETs, which indicates that n -input AND and OR gates consist of $6(n - 1)$ MOS FETs ($n \geq 2$). On the other hand, in fusion gate logic, a n -input AND gate consisted of $4(n - 1)$ MOS FETs, whereas $4n - 2$ MOS FETs were used in an n -input OR gate, as shown in Figs. 4(a) and (b). Therefore, in case of AND logic, the number of MOS FETs in fusion gate circuits is smaller than that of classical circuits. The difference will be

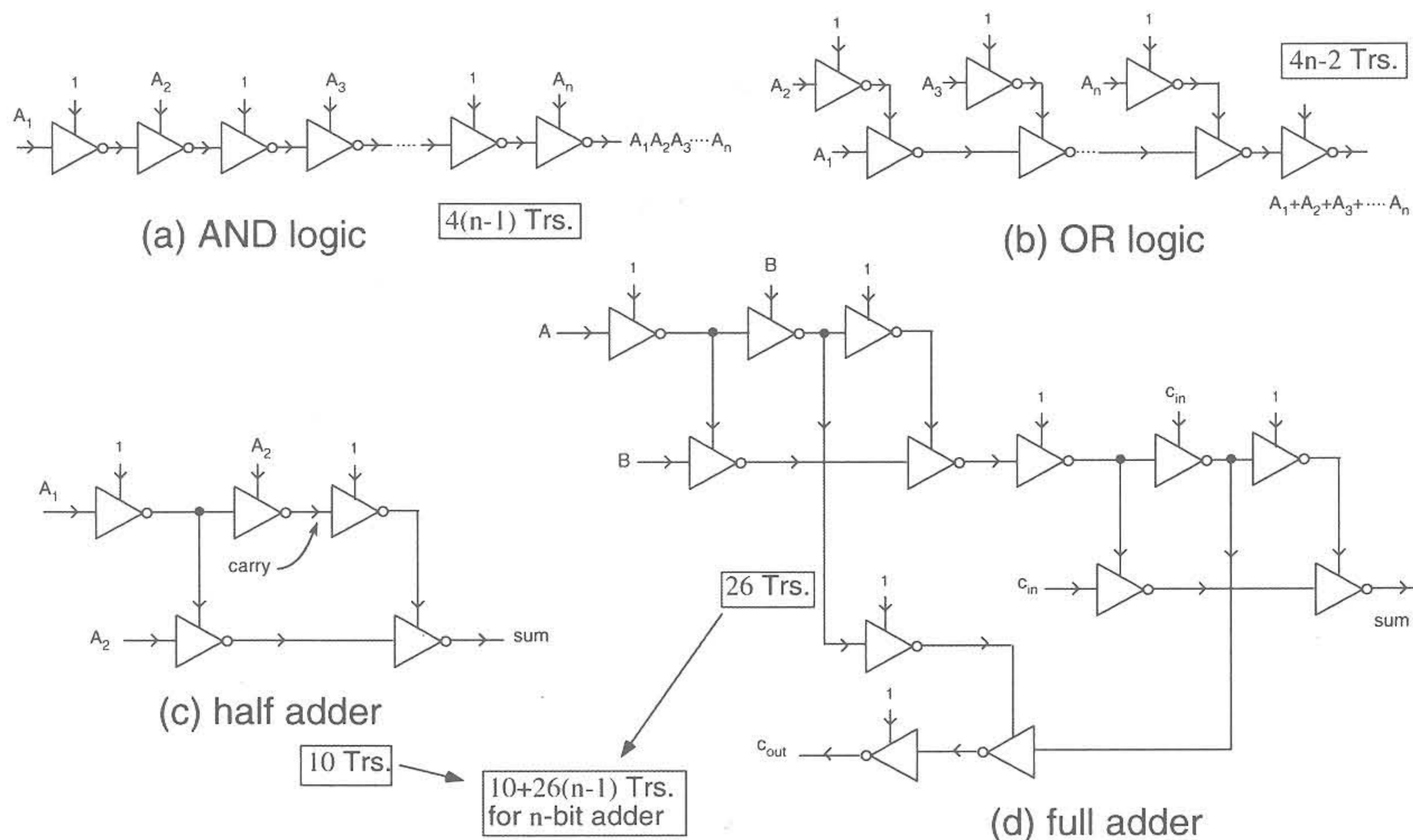


FIGURE 4 Fusion gate architectures of multiple-input functions; (a) AND and (b) OR. Half and full adders are shown in (c) and (d), respectively.

expanded as n increases. Half- and full adders constructed by fusion gate logic are illustrated in Figs. 4(c) and (d). The number of MOS FETs in a classical half adder was 22, while it was 10 in a fusion gate half adder [Fig. 4(c)]. For n -bit full adders ($n \geq 1$), the number of MOS FETs in a classical circuit was $50n - 28$, while it was $26(n - 1) + 10$ in a fusion gate circuit [Fig. 4(d)]. Again, the fusion gate circuit has a significantly smaller number of MOS FETs, and the difference will be increased as n increases.

Figure 5 summarizes the comparison of the number of MOS FETs between classical and fusion gate logic. The number of MOS FETs in fusion gate logic was always smaller than that of MOS FETs in classical logic circuits. Remember that the number of MOS FETs on these circuits' current

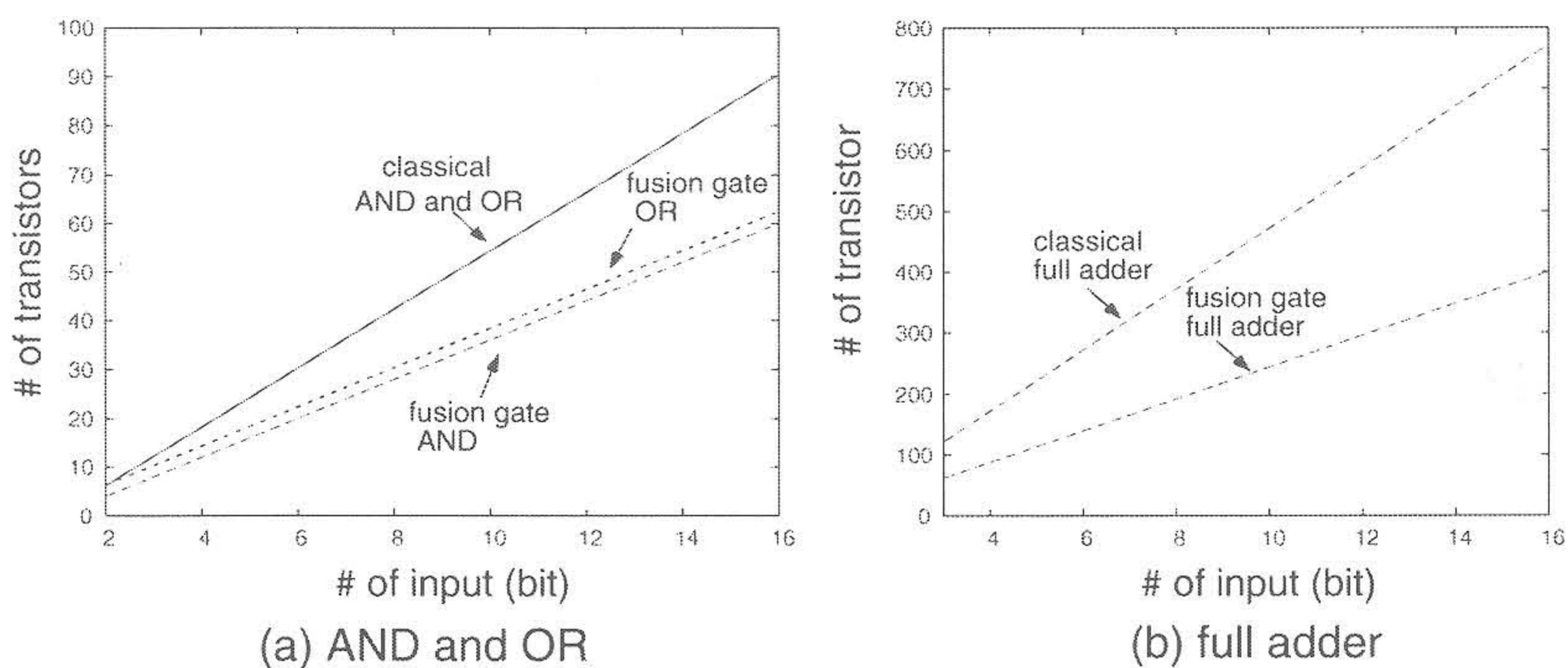


FIGURE 5 Total number of MOS FETs in classical and collision-based multiple-input logic gates.

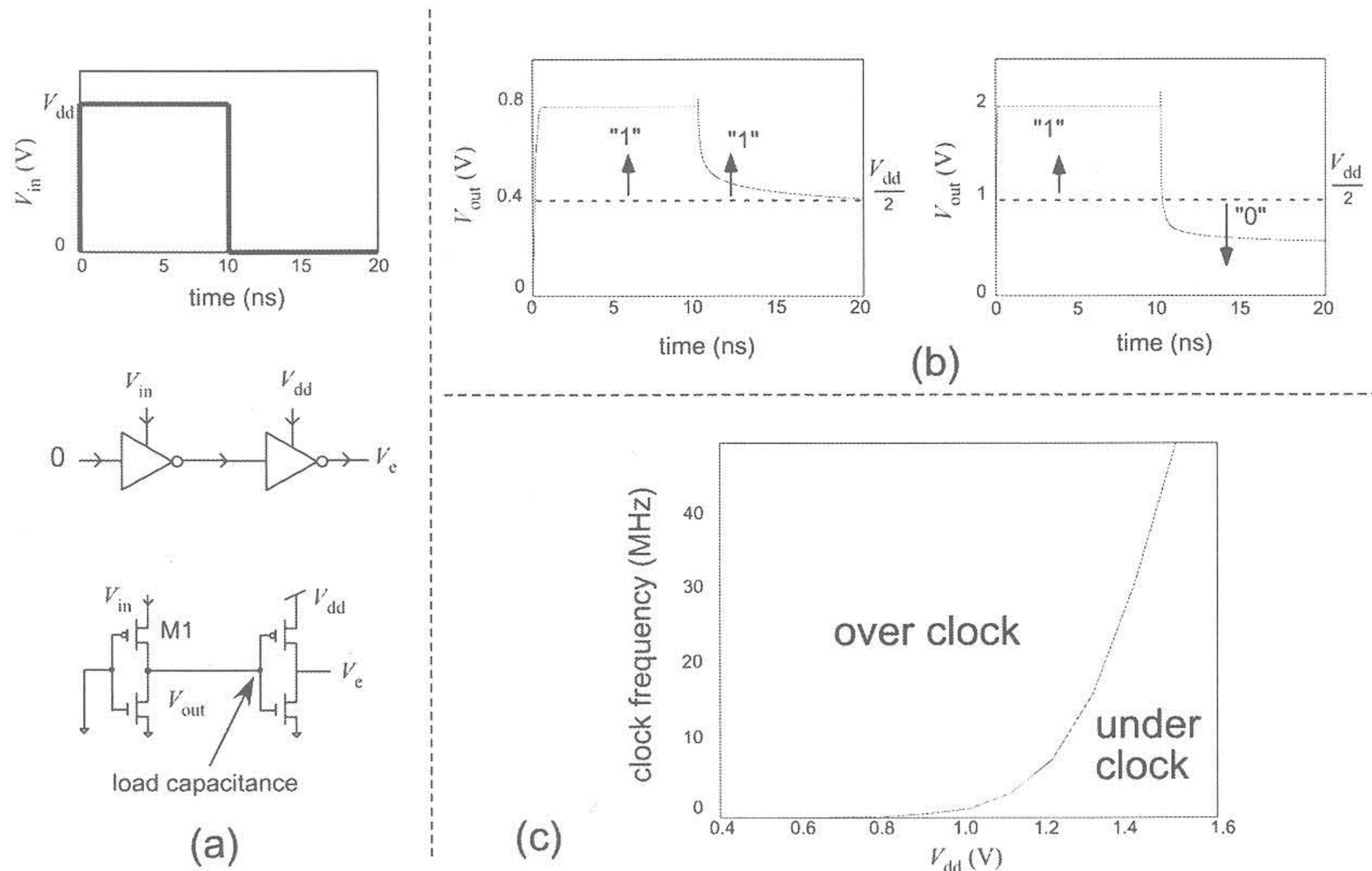


FIGURE 6

Simulation results of upper limit of fusion-gate operation; (a) experimental setup, (b) time courses of outputs of fusion gate with different supply voltages and (c) operation condition with respect to clock frequency and supply voltage.

paths is always smaller than three, which indicates that the supply voltage can be decreased to the same degree as conventional low-power 2-input NAND-based circuits, although the number of MOS FETs in fusion-gate logics is always smaller than that of NAND-based circuits.

A relationship between a clock frequency and a supply voltage is very important for evaluating performances of digital circuits. Figure 6 shows circuit simulation results of fusion gates concerning the operation limit under a given clock frequency and the supply voltage. We used a simulation program with integrated circuit emphasis (SPICE) with $0.35\text{-}\mu\text{m}$ digital CMOS parameters (MOSIS, Vendor TSMC) with minimum-sized transistors. Figure 6(a) shows the experimental setup. A square voltage wave ($V_{in} = 0$ V or V_{dd}) with infinite rising and falling times was given to one input (source of M1) of a fusion gate, whereas the other input was fixed at zero. A load capacitance, which is an input parasitic capacitance of the other fusion gate in next stage, was added to confirm the discharging performance. As shown in Fig. 6(b) left, when V_{dd} was set at 0.8 V and V_{in} was decreased from V_{dd} to zero (at $t = 10$ ns), V_{out} could not reach $V_{dd}/2$ within 10 ns, which indicated that V_{out} was always logical "1", although it must be logical "0" when $t = [10:20]$ ns. In other words, this fusion gate cannot respond to a clock input of 50 MHz ($= 1/20$ ns) when V_{dd} is set at 0.8 V. On the other hand, when V_{dd} was set at 2.0 V, V_{out} reached $V_{dd}/2$ just after V_{in} was decreased from V_{dd} to zero at $t = 10$ ns, as

shown in Fig. 6(b) right. In this case, V_{out} can represent logical “1” within $t = [10:20]$ ns. This means that the fusion gate can respond to a clock input of 50 MHz when V_{dd} is set at 2.0 V. Simulating the circuit under various combinations of input clock frequencies and V_{dds} , we evaluated the upper operation limits as shown in Fig. 6(c). For given combinations, we monitored output node V_e of the load fusion gate and checked whether the output was inverted for given input voltages or not. The upper area in Fig. 6(c) represents the over clock area where the fusion gate circuit cannot respond to the input clocks, whereas the lower area represents the under clock area where the fusion gate circuit can operate correctly.

4 APPLICATIONS: DIGITAL MULTIPLIER AND 4-BIT MICROPROCESSOR

4.1 Digital Multiplier

We designed a digital multiplier using collision-based fusion gates. The multiplier consists of the Booth’s encoder for producing partial products, a data compressor based on the Wallace’s tree for the sum and carry, and a carry-lookahead adder for final product, as shown in Fig. 7; see overviews, e.g. [8].

The Booth’s encoder defines several groups of serial multiple bits and produces partial products for each group (not each bit) for reducing the number of partial products. The circuit for the Booth’s encoder consists of the Booth’s decoder cells (BTDs), sign-bit cells (SIBs) and selector cells (SELS). For M -bit input, the Booth’s encoder requires j ($\equiv M/2$) BTDs, j SIBs and j SELs. A BTD accepts multiplier inputs y_{2j-1} , y_{2j} and y_{2j+1} , and produces the $2j$ -bit shift (Q_1), the $(2j + 1)$ -bit shift (Q_2) and the complement (Q_n), as shown in Fig. 8. The corresponding circuit using 14

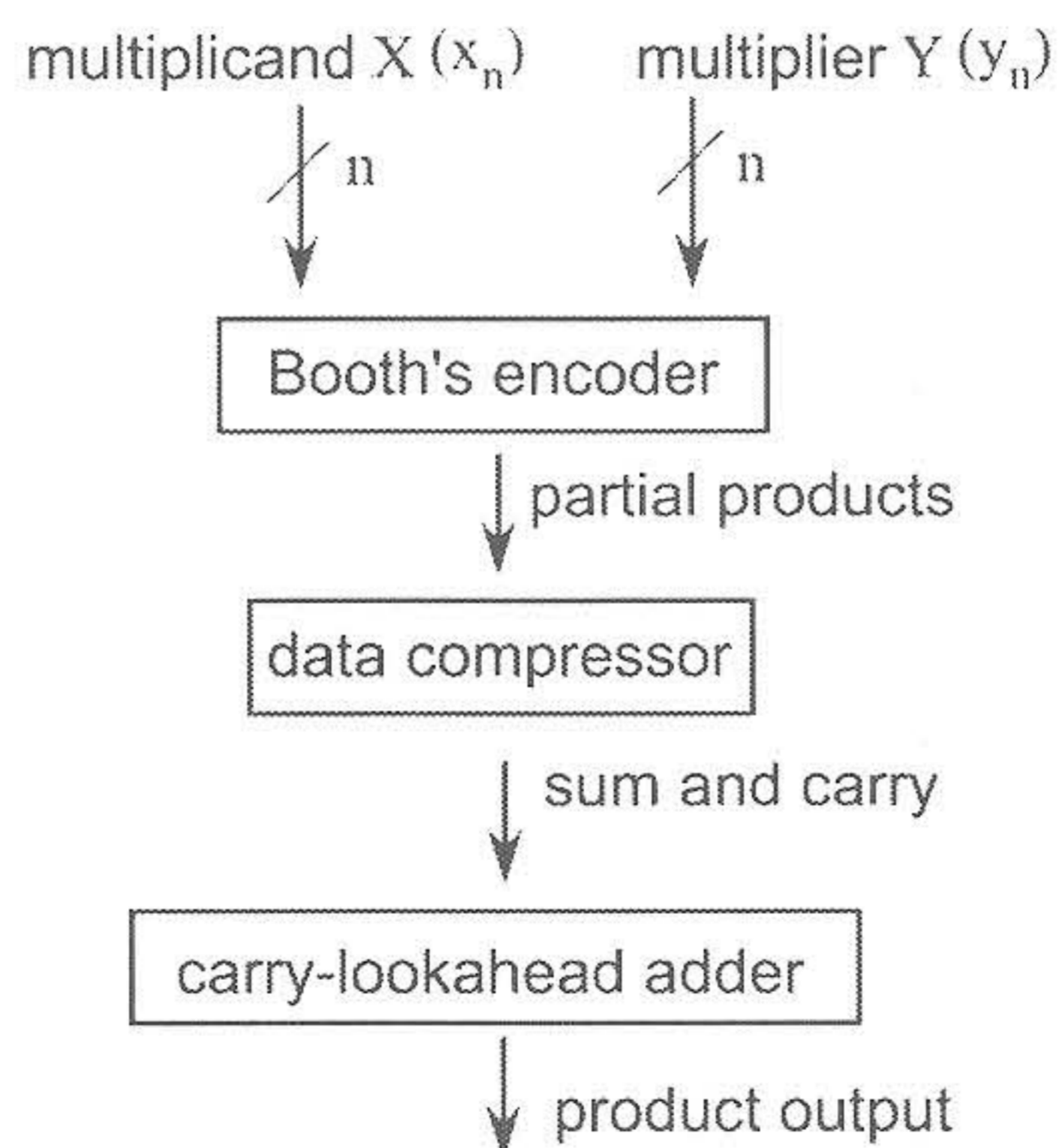


FIGURE 7
Construction of standard digital multiplier.

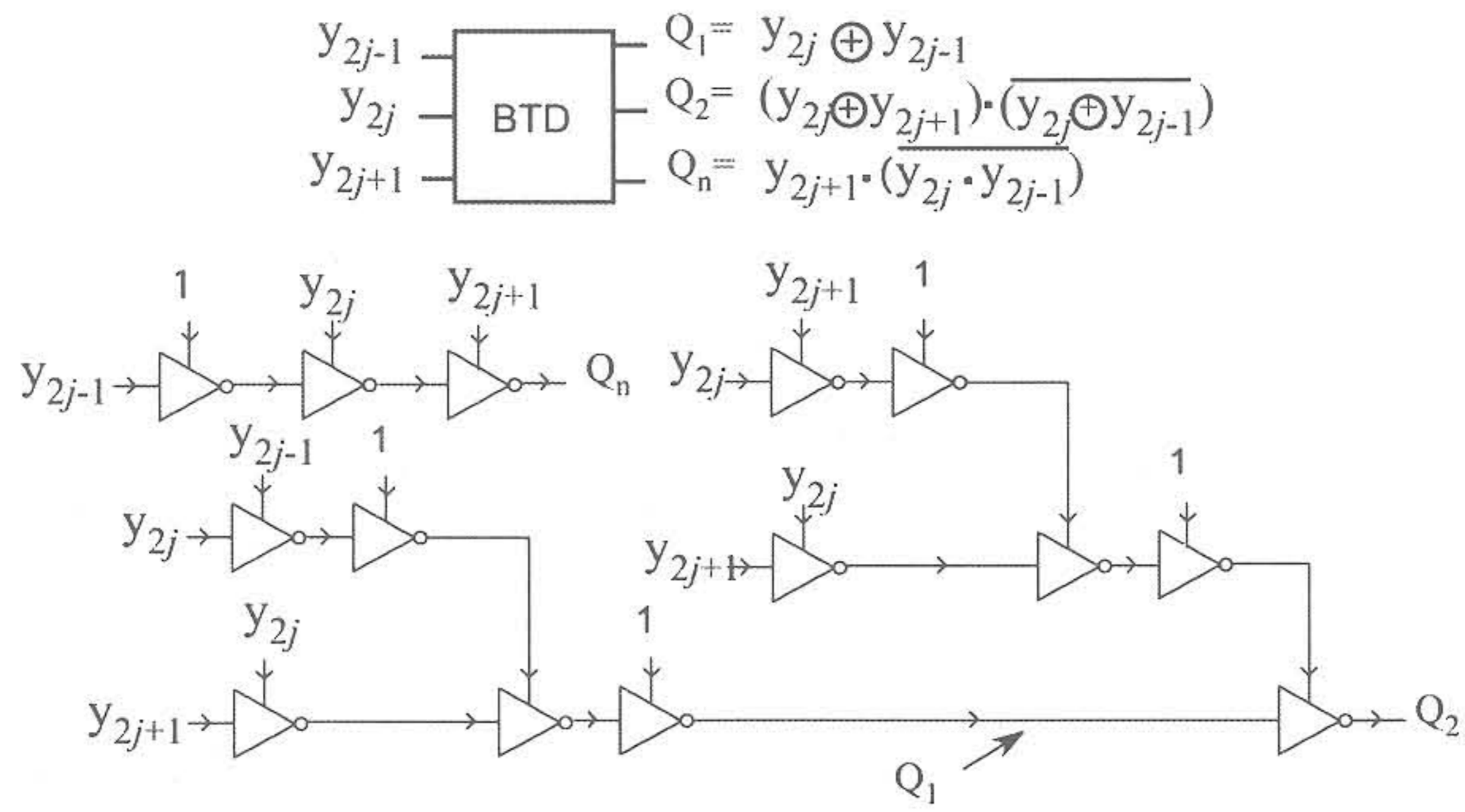


FIGURE 8 Construction of Booth's decoder cell using collision-based fusion gates.

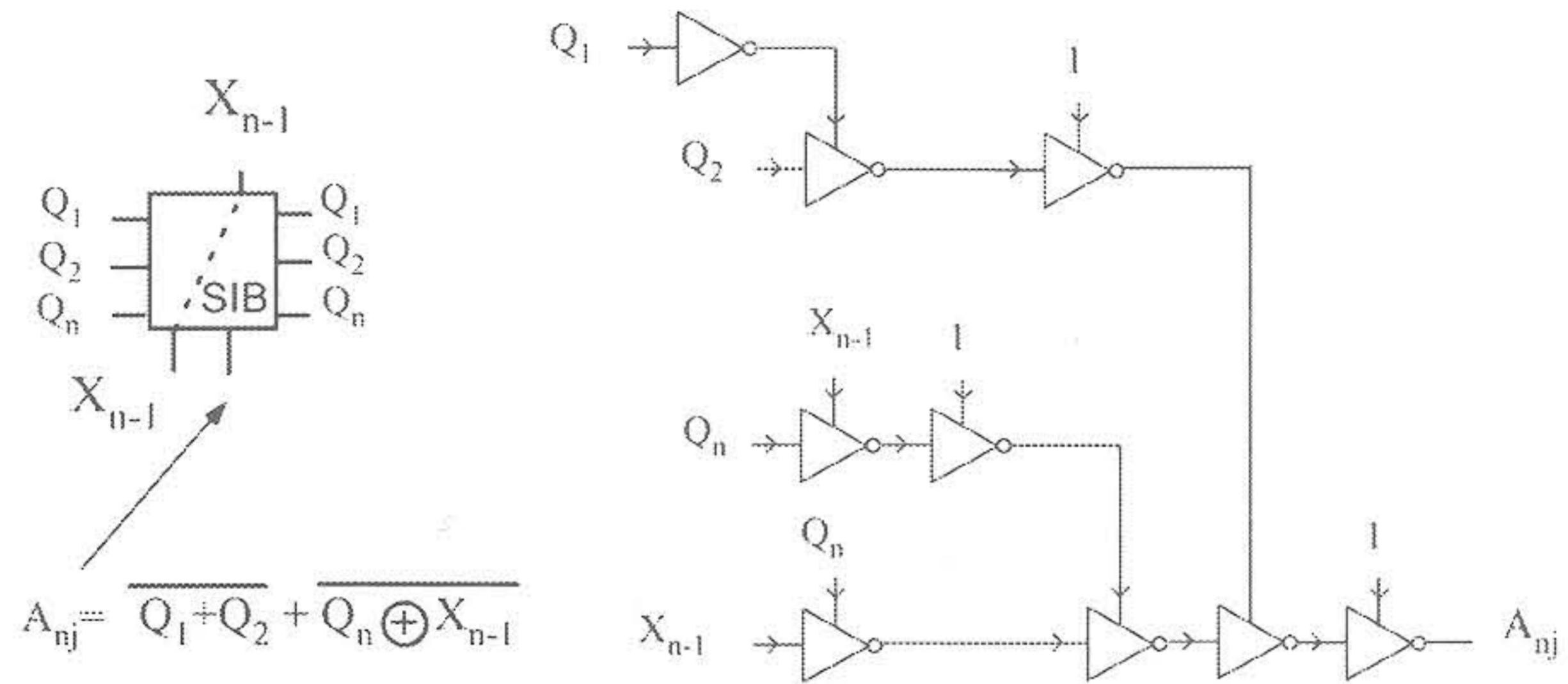


FIGURE 9 Construction of sign-bit cell using collision-based fusion gates.

fusion gates is illustrated in the same figure. A SIB accepts Q_1 , Q_2 and Q_n and produces the sign bit (A_{nj}) with multiplicand bit x_{n-1} (Fig. 9). The circuit consisting of nine fusion gates is shown in the same figure. The number of MOS FETs in BTDs and SIBs for a n -bit multiplier was $46 \times B$ for fusion gates where $B \equiv [n + \text{mod}(n, 2)]/2$, and was $76 \times B$ for conventional gates. Finally, a SEL accepts Q_1 , Q_2 , Q_n and the multiplicand bits (x_{n-1} and x_n), and produces partial product A_{ij} at the j -th stage and the i -th line (Fig.10). The corresponding circuit consisting of ten fusion gates is shown in the same figure. The number of MOS FETs in SELs

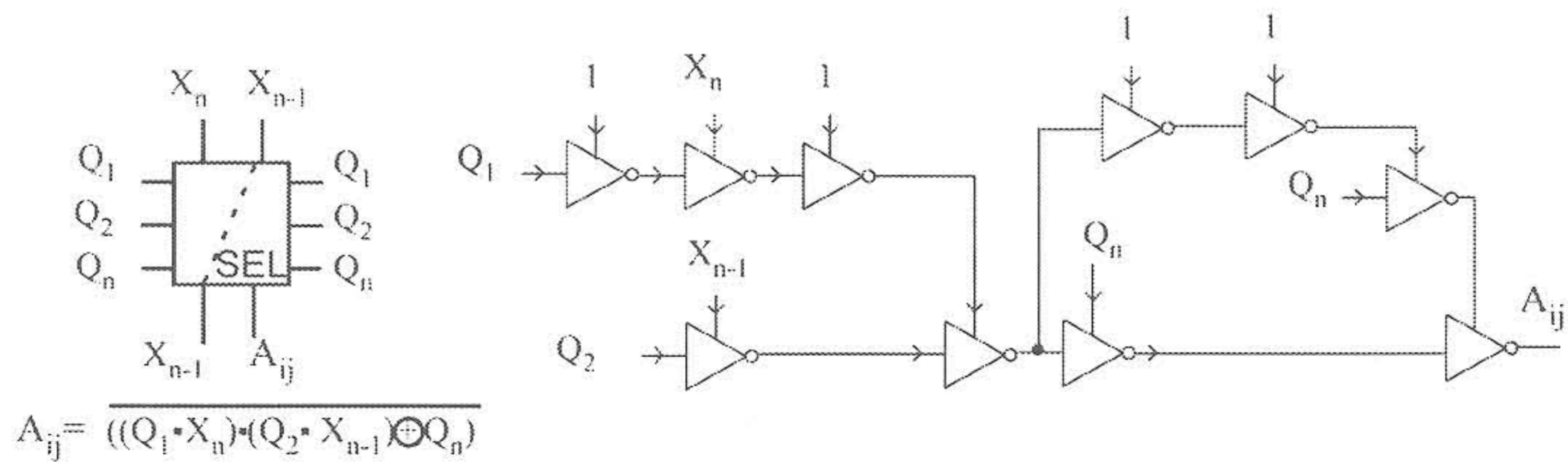


FIGURE 10 Construction of selector cell using collision-based fusion gates.

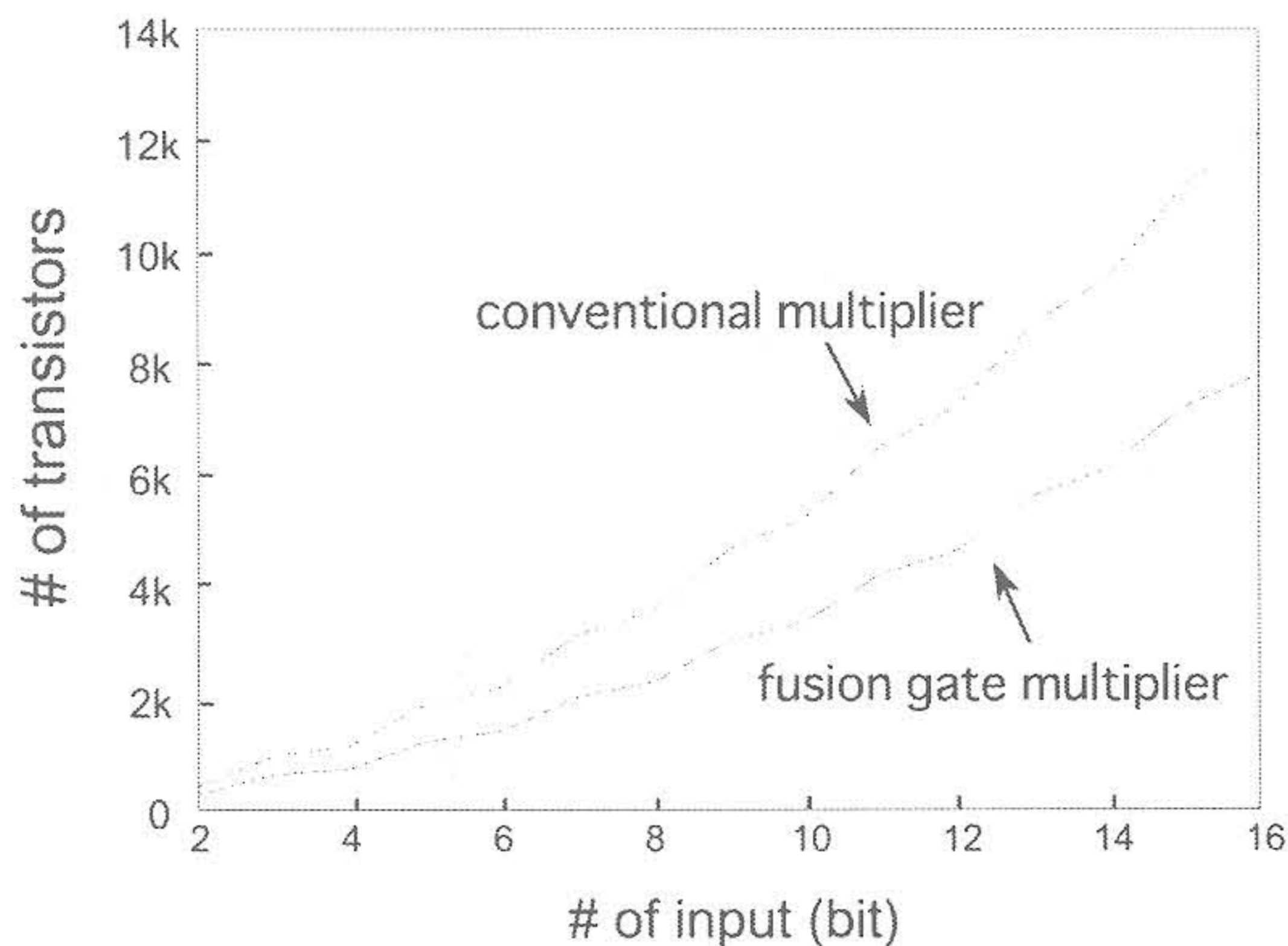


FIGURE 11

Total number of MOS FETs in classical and collision-based n -bit multiplier.

for a n -bit multiplier was $24 \times B$ for fusion gates and was $28 \times B$ for conventional gates.

The data compressor was designed using standard Wallace's tree that reduces the number of partial-product terms by adding partial-product terms having the same bits among all the partial products to multiple three-input two-output full adders step by step. We designed this full adder by using collision-based fusion gates. The number of MOS FETs in the data compressor unit for a n -bit multiplier was $26 \times W$ for fusion gates where $W \equiv n(n/2 - 1)$, and was $50 \times W$ for conventional gates.

Finally, a standard carry-lookahead adder was designed by collision-based AND, OR and XOR gates. The number of MOS FETs in the adder unit for a n -bit multiplier was $180 \times B$ for fusion gates and was $292 \times B$ for conventional gates. Figure 11 summarizes the comparison of the number of MOS FETs between n -bit classical and fusion-gate multiplier. Since the number of MOS FETs is proportional to n and n^2 , this difference will significantly be expanded as n increases.

4.2 4-bit Microprocessor

We designed a 4-bit microprocessor that consists of both combinational and sequential circuits. Although only combinational circuits can be replaced with our fusion gates, we demonstrate that the total number of MOS FETs can significantly be decreased by the replacement. Figure 12 show the block diagram of the 4-bit microprocessor. It has several combinational logic blocks, i.e., a selector, a decoder and an arithmetic logic unit (ALU) consisting of 4-bit collision-based full adder. It also has sequential circuits consisting of three registers (A, B, output and program counter, each of which consists of four D-type flip flops). An additional D-type flip flop receiving a carry flag of the ALU as well as a read-only memory (ROM)

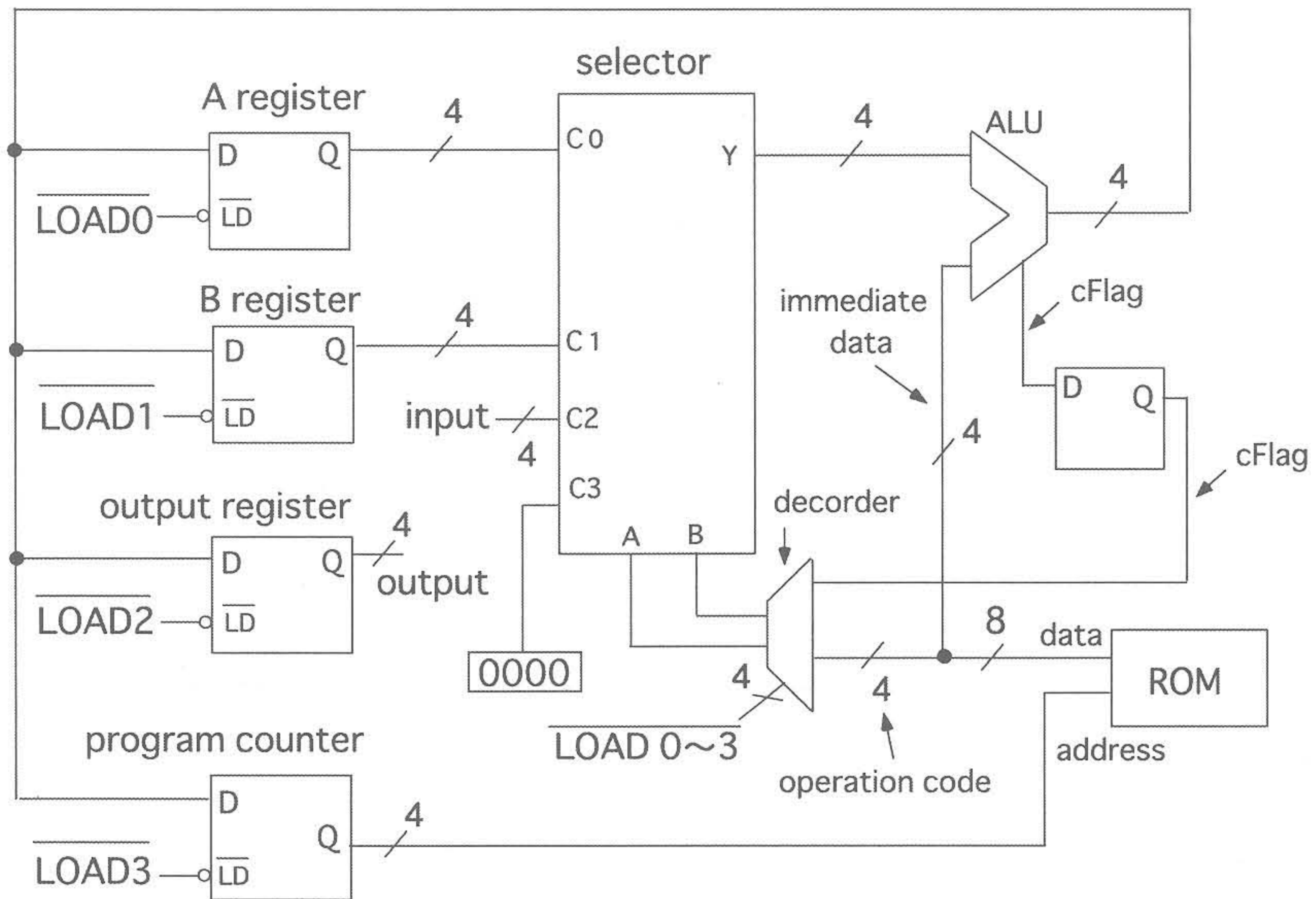


FIGURE 12 Construction of 4-bit microprocessor.

circuit that stores 8-bit instructions consisting of immediate data (4 bit) and operation codes (4-bit) were prepared.

The operation of this microprocessor is very simple. First, the program counter is counted up when $LOAD3 = "1"$. The ROM address is selected by the program counter step by step, and instructions are then readout from the ROM circuit. The 4-bit operation code is given to the decoder. At the same time, the 4-bit immediate data is given to the ALU. The selector accepts outputs of A and B registers (C0 and C1), external input data (C2), zero (C3) and the decoder output (A and B). The ALU accepts the outputs of the selector ($Y = C0$ or $C1$ or $C2$ or $C3$ selected by decoder outputs A and B). Note that the zero input (C3) is necessary for avoiding unwanted summations in the ALU in the case of data transmission. A carry flag (cFlag) generated by the ALU is stored in a D-type flip flop. When $cFlag = "1"$, the decoder must produce no operation code (NOP) because the data cannot be handled by this microprocessor.

Table 1 shows 17 operation codes of this microprocessor and the corresponding inputs and outputs of the decoder. The 4-bit operation code is represented by OP1, OP2, OP3 and OP4. In the table, "x" represents 'does not matter' and "Im" represents the immediate data stored in the ROM circuit. Instruction "ADD A, Im" adds immediate data Im to A register. "MOV A, B" moves data in B register to A register. "IN A" stores external input data to A register, while "OUT B" stores data in B register to the output register. "JNC Im" is a conditional jump to the address stored in

command	input					output					
	OP 3	OP 2	OP 1	OP 0	cFlag	A	B	LOAD 0	LOAD 1	LOAD 2	LOAD 3
ADD A, Im	0	0	0	0	X	0	0	0	1	1	1
MOV A,B	0	0	0	1	X	0	1	0	1	1	1
IN A	0	0	1	0	X	1	0	0	1	1	1
MOV A,Im	0	0	1	1	X	1	1	0	1	1	1
MOV B,A	0	1	0	0	X	0	0	1	0	1	1
ADD B,Im	0	1	0	1	X	0	1	1	0	1	1
IN B	0	1	1	0	X	1	0	1	0	1	1
MOV B,Im	0	1	1	1	X	1	1	1	0	1	1
OUT B	1	0	0	1	X	0	1	1	1	0	1
OUT Im	1	0	1	1	X	1	1	1	1	0	1
JNC Im	1	1	1	0	0	1	1	1	1	1	0
NOP	1	1	1	0	1	X	X	1	1	1	1
JMP Im	1	1	1	1	X	1	1	1	1	1	0

TABLE 1
Command sets of 4-bit microprocessor.

immediate data Im when cFlag = "0". "NOP" represents the no operation because cFlag = "1" as explained above. "JMP Im" is a normal jump code to address in immediate data Im.

Figure 13 shows the constructions of the 4-bit selector consisting of 68 fusion gates (17 fusion gates for 1-bit selector). Figure 14 shows the fusion gate implementation of the decoder and ALU. The decoder consisted of ten fusion gates, whereas the ALU was constructed by four collision-based full

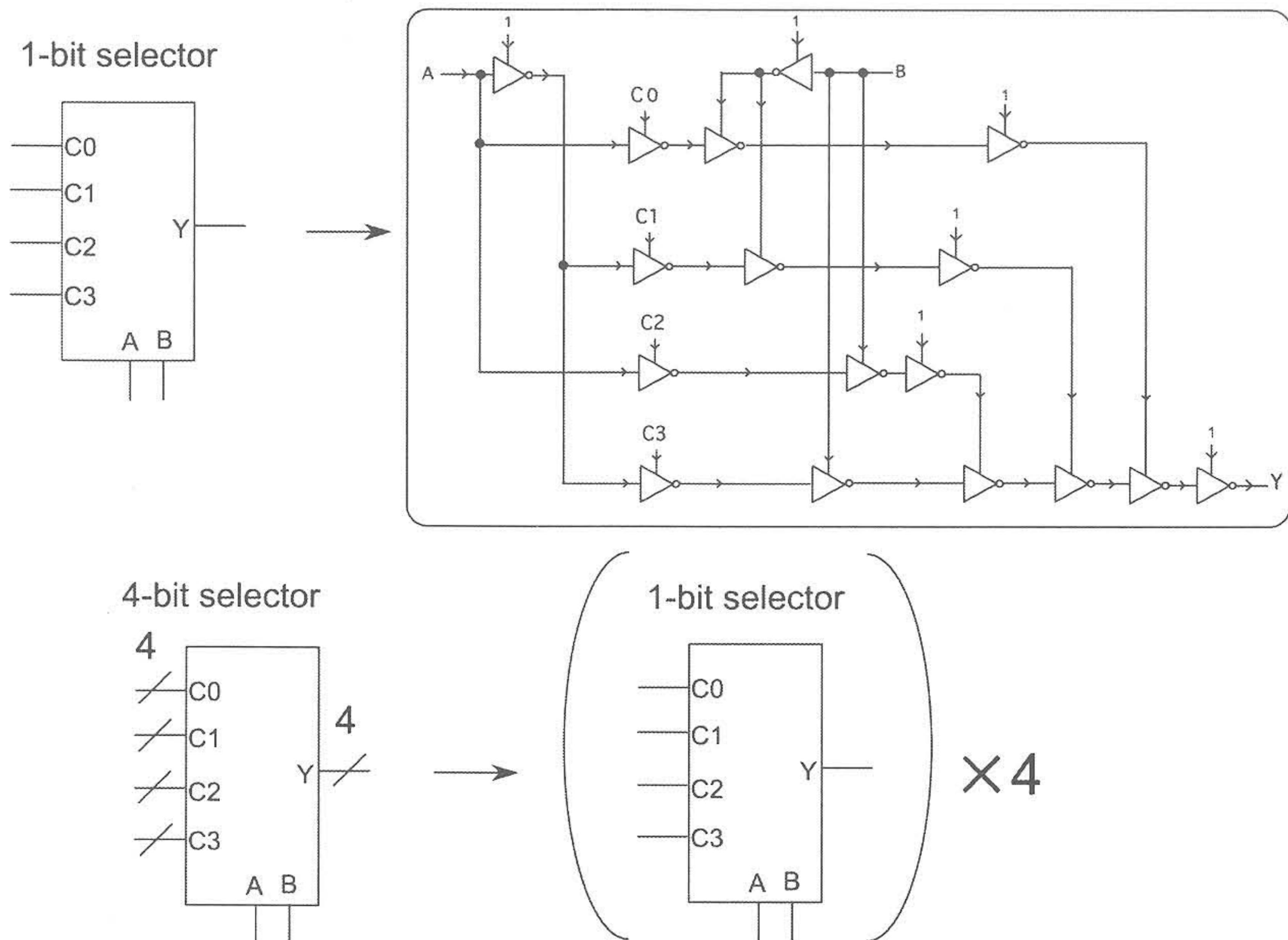


FIGURE 13
Construction of 1-bit and 4-bit selector circuits using fusion gates.

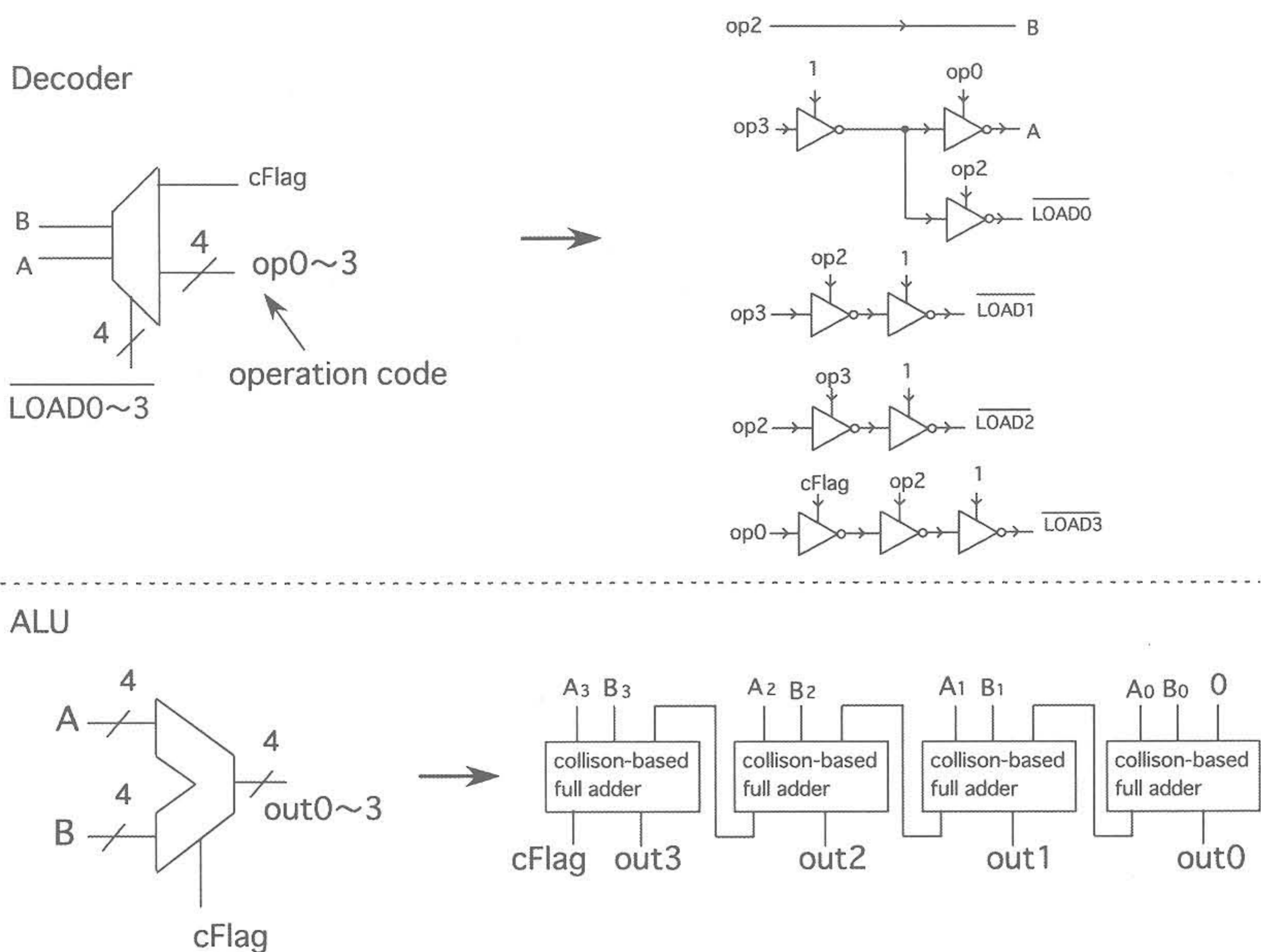


FIGURE 14 Construction of decoder and ALU circuits using fusion gates.

adders. The total number of MOS FETs in the collision-based microprocessor was 920, while it was 1276 when conventional AND-based logic circuits were used.

It should be remembered that the number of MOS FETs on these circuits' current paths is always smaller than three. Therefore, the supply voltage can be decreased to the same degree as conventional low-power 2-input NAND-based circuits. Indeed, when two inputs of a fusion gate were "0", the output voltage is not completely zero because of the threshold voltage of the MOS FETs. This small voltage shift can be restored to logical "0" at the next input stage with the three-transistor constraints. Therefore, additional level-restoring circuits are unnecessary for our fusion gate circuits.

5 SUMMARY

We described a method of designing logic circuits inspired by collision-based reaction-diffusion computing. First, we introduced a new interpretation of collision-based computing, especially concerning a limited direction of wave fragments and infinite transition speed. This simplified constructions of the computing media significantly. Second, we showed that basic logical functions were able to be represented in terms of our unit operator, i.e., a 'fusion gate', that calculated \overline{AB} or $A\overline{B}$ for inputs A and B. Third,

basic MOS circuits for the fusion gate that consisted of two MOS FETs were introduced. Then we demonstrated that in case of multiple-input logic functions and a digital multiplier, the number of MOS FETs in fusion gate circuits was smaller than that of classical circuits, and the difference will significantly be expanded as n increases. Finally, we implemented a 4-bit microprocessor as a practical example. Since the combination of the fusion gates produces multiple functions, e.g., an AND circuit can compute NOR simultaneously, we should build optimization theories for generating multiple-input arbitrary functions.

ACKNOWLEDGMENTS

The authors wish to thank Professor Andrew Adamatzky of the University of the West of England for valuable discussions and suggestions during the research, and Professor Masayuki Ikebe of Hokkaido University for suggestions concerning various CMOS circuits. This study was partly supported by Industrial Technology Research Grant Program in '04 from New Energy and Industrial Technology Development Organization (NEDO) of Japan, and a Grant-in-Aid for Young Scientists [(B)17760269] from the Ministry of Education, Culture Sports, Science and Technology (MEXT) of Japan.

REFERENCES

- [1] Adamatzky, A. (2002). Editor, *Collision-Based Computing*, Springer-Verlag.
- [2] Adamatzky, A. (2004). Computing with waves in chemical media: Massively parallel reaction-diffusion processors, *IEICE Trans. Electron.*, *E87-C*, (11), 1748–1756.
- [3] Alioto, M. and Palumbo, G. (2005). Model and Design of Bipolar and MOS Current-Mode Logic: CML, ECL and SCL Digital Circuits, Springer.
- [4] Asahi, N., Akazawa, M., and Amemiya, Y. (1998). Single-electron logic systems based on the binary decision diagram, *IEICE Trans. Electronics*, *E81-C*, (1), 49–56.
- [5] Berlekamp, E.R., Conway, J.H., and Guy, R.L. (1982). *Winning Ways for your Mathematical Plays*. 2. Academic Press.
- [6] Fredkin, F. and Toffoli, T. (1982). Conservative logic, *Int. J. Theor. Phys.*, *21*, 219–253.
- [7] Margolus, N. (1984). Physics-like models of computation, *Physica D*, *10*, 81–95.
- [8] Lee, H. (2005). Power-Aware Scalable Pipelined Booth Multiplier, *IEICE Trans. Fund.*, *E88-A*, (11), 3230–3234.
- [9] Motoike, I.N. and Yoshikawa, K. (2003). Information operations with multiple pulses on an excitable field, *Chaos, Solitons and Fractals*, *17*, 455–461.
- [10] Motoike, I. and Yoshikawa, K. (1999). Information operations with an excitable field, *Phy. Rev. E*, *59*, (5), 5354–5360.
- [11] Muller, D.E. (1954). Application of Boolean Algebra to Switching Circuit Design and to Error Detection, *IRE Trans. on Electr. Comp.*, *EC-3*, 6–12.
- [12] Reed, I.S. (1954). A Class of Multiple-Error-Correcting Codes and Their Decoding Scheme, *IRE Trans. on Inform. Th.*, *PGIT-4*, 38–49.

- [13] Shelar, R.S. and Sapatnekar, S.S. (2001). BDD decomposition for the synthesis of high performance PTL circuits, *Workshop Notes of IEEE IWLS*, 298–303.
- [14] Soeleman, H. and Roy, K. (1999). Ultra-low power digital subthreshold logic circuits, in *Proc. Int. Symp. on low power electronics and design*, 94–96.
- [15] Soeleman, H., Roy, K., and Paul, B.C. (2001). Robust subthreshold logic for ultra-low power operation, *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 9, (1), 90–99.
- [16] Song, M. and Asada, K. (1998). Design of low power digital VLSI circuits based on a novel pass-transistor logic, *IEICE Trans. Electronics, E81-C*, (11), 1740–1749.
- [17] Steiglitz, K., Kamal, I., and Watson, A. (1988). Embedded computation in one-dimensional automata by phase coding solitons, *IEEE Trans. Comp.*, 37, 138–145.
- [18] Yamada, T., Kinoshita, Y., Kasai, S., Hasegawa, H., and Amemiya, Y. (2001). Quantum-dot logic circuits based on the shared binary decision diagram, *Jpn. J. Appl. Phys.*, 40, (7), 4485–4488.