Physica C 468 (2008) 1983–1986

Contents lists available at ScienceDirect

Physica C

journal homepage: www.elsevier.com/locate/physc

Single-flux-quantum logic circuits exploiting collision-based fusion gates

T. Asai *, K. Yamada, Y. Amemiya

Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Japan

ARTICLE INFO

Article history:
Available online 13 June 2008

PACS:
02.70
82.25.H

Keywords:
Single-flux-quantum
Logic computation
Reaction–diffusion computers

ABSTRACT

We propose a single-flux-quantum (SFQ) logic circuit based on the fusion computing systems––collision-based and reaction–diffusion fusion computers. A fusion computing system consists of regularly arrayed unit cells (fusion gates), where each unit has two input arms and two output arms and is connected to its neighboring cells with the arms. We designed functional SFQ circuits that implemented the fusion computation. The unit cell was able to be made with ten Josephson junctions. Circuit simulation with standard Nb/Al–AlOx/Nb 2.5-kA/cm^2 process parameters showed that the SFQ fusion computing systems could operate at 10 GHz clock.

1. Introduction

Collision-based computing, which was originally introduced in conservative computation such as a billiard-ball model [1] and its cellular-automaton (CA) analogues [2], presents a novel approach of computation with mobile physical objects (e.g., billiard-balls), chemical particles, self-localized patterns on CAs, and so on. Although a number of mathematical and abstract models have been proposed [3], the physical (LSI) implementation is still difficult because (i) collision-based physical models require conservative media for series computation and (ii) CA systems have to implement complex rules per a cell for producing self-localized patterns. Collision-based fusion gates, which fuse billiard-ball models and excitable properties of chemical waves in reaction–diffusion media [4], have recently been proposed as a possible solution to LSI implementation of collision-based computers [3,5,6]. The fusion computing is a way of digital processing that performs logic operation, and is very powerful for implementing a given digital operation with a small number of logic gates [6]. In this work, we use single-flux-quantum (SFQ) circuits to construct fusion computing circuits for combinational logic processing.

In collision-based computing, information propagates in an impulse form; i.e., existence of a mobile object, particle or self-localization is represented by a spatial impulse on the abstract media. Unlike other electronic devices, a medium for signals in SFQ circuits is a pulse of a fluxoid quantum, therefore SFQ circuits are able to implement the collision-based computers more simply. In the following sections, we first outline the fusion computation and describe the fusion gates consisting of SFQ circuits (Section 2), and then we describe its circuit implementation and demonstrate the operation of fundamental logic gates by means of computer simulation (Section 3).

2. Fusion computation with SFQ circuits

2.1. Logic computation with fusion gates

A fusion computation system consists of regularly arrayed unit cells (fusion gates) where each cell has two input arms and two output arms and is connected to its neighboring cells with the arms (Fig. 1). A unit cell accepts two logic inputs from its North and South arms, and outputs two logic signals to the East and South arms. The cell's inputs are represented by logical variables A and B where A (or B) = '1' represents the existence of an impulse train traveling North–South (or West–East), and A (or B) = '0' represents the absence of impulse trains. When A = B = '1' we assume that impulses collide at the center position (black circle) and then disappear, as observed in excitable chemical waves on reaction–diffusion media [2,4]. East and South outputs are thus '0' because of the disappearance. If A = B = '0', the outputs will be '0' as well because of the absence of the impulses. When A = '1' and B = '0', an impulse given at the North node (A) can travel to the South because it does not collide with an impulse 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, output logical functions of this simple 'gate' are represented by $\overline{A}B$ and $A\overline{B}$($\overline{A}$ and $\overline{B}$ represent logical negation of A and

* Corresponding author. Tel.: +81 11 706 6080; fax: +81 11 706 7890.
E-mail address: asai@sapiens-ei.eng.hokudai.ac.jp (T. Asai).

0921-4534/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.physc.2008.05.273

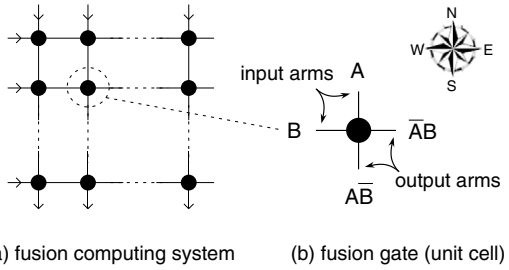

Fig. 1. Fusion computing system: (a) 2D array of fusion gates and (b) unit function of single fusion gate (unit cell).

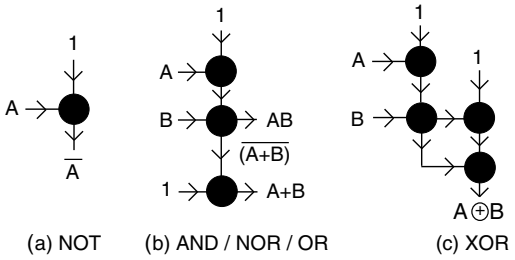

Fig. 2. Examples of logic functions composed of fusion gates: (a) NOT function with one fusion gate, (b) AND/NOR/OR functions with three gates and (c) XOR function with four gates.

B, respectively). We call this unit a 'fusion gate' that utilizes properties of collision-based kinetics and excitable waves.

Fig. 2 represents basic logic circuits constructed by combining several fusion gates. The simplest example is shown in Fig. 2a where the NOT function is implemented by a single 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 ($\overline{A}$). Fig. 2b represents a combinational circuit of three fusion gates that produces AND, NOR, and OR functions simultaneously. Exclusive logic functions (XOR and XNOR) are constructed by three fusion gates as shown in Fig. 2c. Any combinational circuit can be constructed by combining fusion gates [5,6].

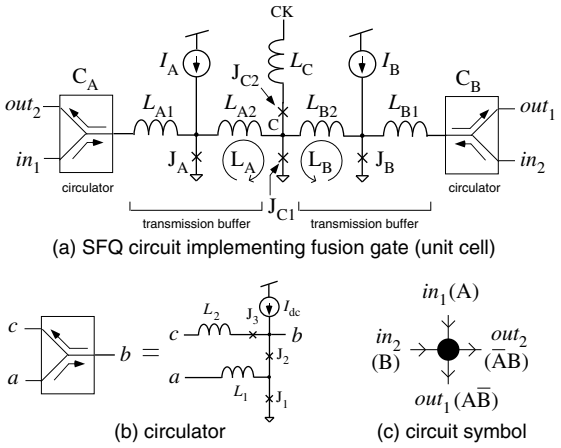

Fig. 3. SFQ circuit of fusion gate: (a) entire circuit consisting of two circulators (C_A and C_B), two transmission buffers and two Josephson memory loops (L_A and L_B), (b) construction of circulator and (c) circuit symbol of SFQ fusion gate.

2.2. Fusion device consisting of SFQ circuits

We designed functional SFQ circuits that implemented the fusion gate. Fig. 3 shows the unit cell consisting of two circulator circuits and two transmission buffers including two memory loops (L_A and L_B).

A circulator consists of three Josephson junctions J_1–J_3, and two inductances L_1 and L_2 biased by a constant current I_{dc}, as shown in Fig. 3b. Let I_i and I_{sfq} denote the critical current of J_i and the loop current of an SFQ, respectively. When (i) $I_1 < I_{sfq} + I_{dc}$, $I_3 > I_{sfq}$ and (ii) $I_2 < I_{sfq} + I_{dc}$, an input SFQ pulse given to node '*a*' is transmitted to node '*b*' because J_1 (and then J_3) is switched to a resistive state. On the other hand, a pulse given to node '*b*' is not transmitted to node '*a*' because J_2 (and then J_3) is switched to a resistive state. An input SFQ pulse given to node '*a*' is not transmitted to node '*c*' because the SFQ current of J_2 branches to J_3 and node '*b*', and the branched current to J_3 is faded away when J_3 is switched to a resistive state. Contrary, an input SFQ pulse given to node '*b*' is transmitted to node '*c*' with this condition because J_2 is switched to a resistive state and the SFQ current is directly flowing into J_3

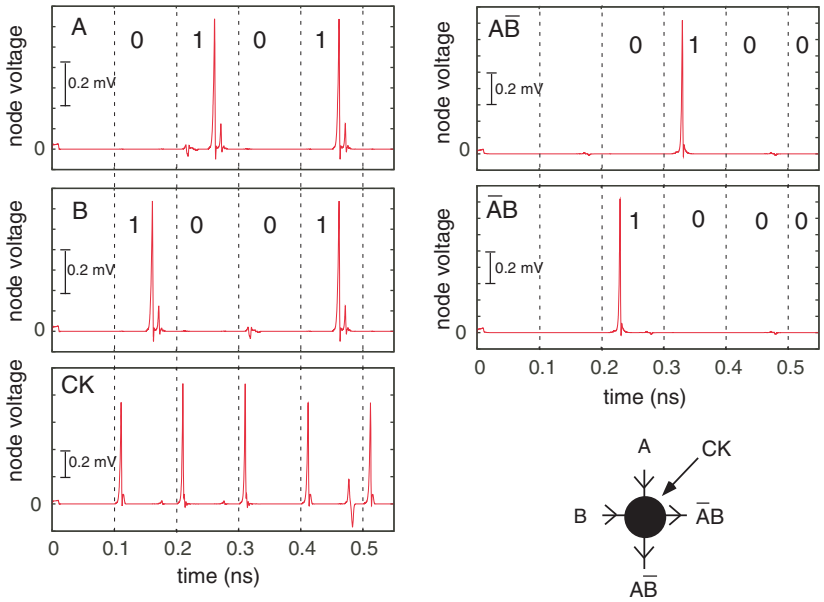

Fig. 4. Simulation results of SFQ fusion gate (transient voltage waveforms).

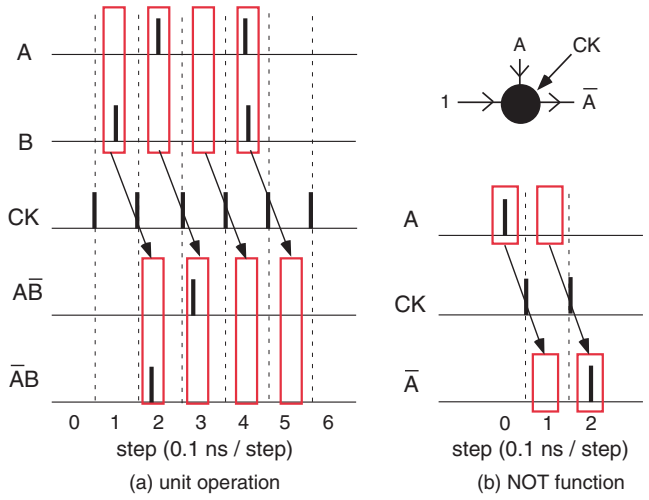

Fig. 5. Impulse raster plots of (a) single SFQ gate and (b) SFQ NOT circuit obtained from circuit simulations.

(J_3 is then switched to a resistive state by this non-branched current). Therefore this circuit transmits flux quanta from node '*a*' to '*b*' and from '*b*' to '*c*' only.

In Fig. 3a, when a SFQ pulse is applied to node in_1 only (A = '1' and B = '0'), it propagates to center node C via circulator C_A and a transmission buffer (L_{A1}, L_{A2}, J_A and I_A). When clock node CK was set at '0', the SFQ is stored in Josephson memory loop L_A. When a voltage clock pulse is applied to CK, the stored SFQ is transmitted to circulator C_B and is sent to out_1. It should be noted that this processing requires one clock step. Similarly, when A = '0' and B = '1' (a SFQ pulse is applied to node in_2 only), the input SFQ is transmitted to out_2 only. When A = '1' and B = '1', the sum of SFQ currents in loops L_A and L_B exceed the critical current of J_{C1}. Therefore the SFQs in L_A and L_B fade away, and no SFQs are transmitted to out_1 and out_2. When A = '0' and B = '0', no SFQ outputs are generated at out_1 and out_2. Consequently, representing the existence and absence of SFQ pulses at input and output nodes by logical variables, this circuit computes $\bar{A}B$ and $A\bar{B}$ from inputs A and B. Fig. 3c illustrates the circuit symbol and the correspondence between the circuit nodes and the input–output logics.

3. Construction of SFQ fusion computing circuits and simulation results

We confirmed operations of SFQ fusion gates through a computer simulation assuming a standard Nb/Al–AlOx/Nb 2.5-kA/cm^2 process. In the simulations, all the inductances (L_1, L_2, L_{A1}, L_{A2}, L_{B1}, L_{B2} and L_C) and bias currents (I_{dc}, I_A and I_B) were set at 8 pH and 0.1 mA. Critical currents of Josephson junctions (J_1, J_2, J_3, J_A, J_B, J_{C1} and J_{C2}) were set at (I_1, I_2, I_3, I_A, I_B, I_{C1} and I_{C2}) = (0.2, 0.14, 0.05, 0.14, 0.14, 0.14 and 0.05) mA. As clock signal CK, we applied voltage pulses of 10 GHz (amplitude: 0.5 mV, pulse width: 3 ps).

Fig. 4 shows transient voltage waveforms of a single fusion gate, and the simulated SFQ impulses in Fig. 4 are re-plotted in Fig. 5a for interpretative use. At step 1 [(A, B) = ('0', '1')] in Fig. 5a, memory loop L_B stored the flux quanta from B = '1'. The stored flux quanta were transmitted to node $\bar{A}B$ when clock pulse was applied to CK at the beginning of step 2. Therefore ($\bar{A}B$, $\bar{A}\bar{B}$) = ('0', '1') was obtained at step 2. Remember that this logic processing is delayed by 1 step per a gate. Similarly, (A, B) was set at ('1', '0') at step 2, and then we obtained ($A\bar{B}$, $\bar{A}B$) = ('1', '0') at step 3. At step 3, (A, B) was set at ('0', '0'), and then ($A\bar{B}$, $\bar{A}B$) = ('0', '0') was obtained at step 4 because no SFQ pulses were given. When (A, B) = ('1', '1') (step 4), the flux quanta stored in memory loops L_A and L_B faded away, and therefore ($A\bar{B}$, $\bar{A}B$) was ('0', '0') at step 5. These results indicated that the proposed circuit was able to compute the required logic functions ($A\bar{B}$ and $\bar{A}B$) and could operate at 10 GHz clock when we assumed a standard Nb/Al–AlOx/Nb 2.5-kA/cm^2 process. Fig. 5b shows the extended example of the SFQ fusion gate––the NOT circuit. The operation is the same as in Fig. 5a if

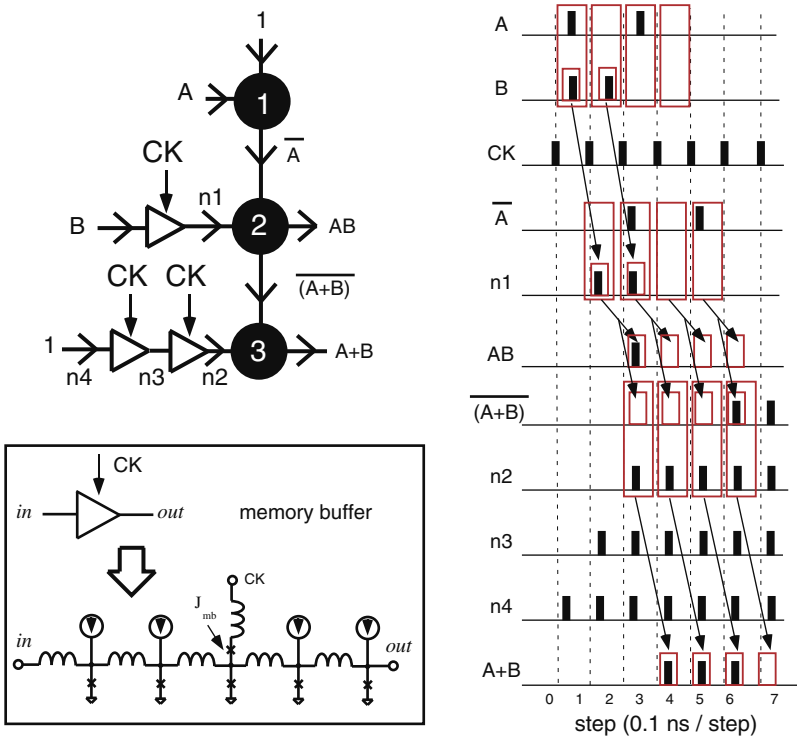

Fig. 6. Raster plots of SFQ AND/NOR/OR circuit obtained from circuit simulations. Triangle symbols represent memory buffer circuit (in solid box).

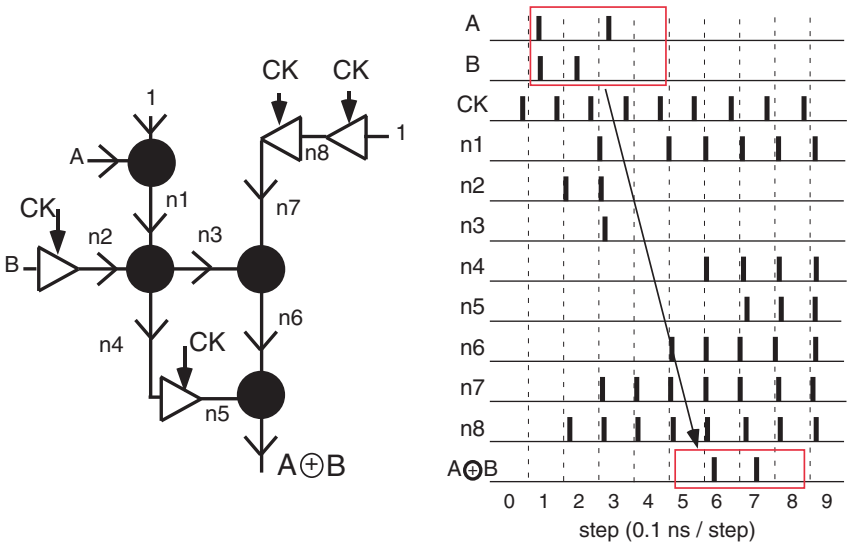

Fig. 7. Raster plots of SFQ XOR circuit obtained from circuit simulations.

we see (A, B) at steps 1 and 4 only, and observe the outputs $\overline{A}B$ at steps 2 and 5. Because B was always '1' in the NOT circuit, we obtained $\overline{A}$ at output node $\overline{A}B$.

Fig. 6 shows a SFQ AND/NOR/OR circuit consisting of three SFQ fusion gates and three memory buffers (triangle symbols in the figure). Because SFQ impulses are delayed by 1 step per fusion gate, series computation using multiple fusion gates with multiple fanout paths requires the timing control of input SFQ pulses. A memory buffer, illustrated in solid box in Fig. 6, is used for this purpose. It stores input flux quanta given at node 'in' in the memory loop, and the stored flux is transmitted to the output node ('out') when external voltage clock CK is given. In the simulation, all the inductances and bias currents in the memory buffer were set at 8 pH and 0.1 mA. Critical currents of Josephson junctions were set at 0.14 mA except for J_{mb} (I_{mb} = 0.05 mA). At step 1, SFQ gate 1 (the gate number is written inside the gate's symbol in the figure) accepted input A and generated $\overline{A}$ at step 2. SFQ gate 2 must accept input n1 and $\overline{A}$ simultaneously. To this end, a memory buffer was inserted between input node B and n1. At step 3, SFQ gate generated AB and $\overline{A+B}$ outputs. Again, SFQ gate 3 must accept $\overline{A+B}$ and '1' simultaneously. Therefore, two memory buffers were inserted between node n4 and n2, and SFQ gate accepted inputs of '1' and (A + B) at step 3. Then the output (A + B) was obtained at step 4. It should be noticed that the output (A + B) was obtained 3 steps later after the inputs (A and B) were given because this circuit used three SFQ gates. Fig. 7 shows a SFQ circuit implementing XOR functions using four SFQ fusion gates and four memory buffers. The XOR outputs were obtained four steps later after the inputs were given, as expected.

The present SFQ fusion gate requires clocks and memory buffers for the timing control. This means, if the number of fanout paths were increased, the time required for obtaining the final results is also increased. A possible solution to this problem is to eliminate the timing control circuitry and to use multiple SFQ impulses (impulse trains) to represent the input and output logic values, instead of a single SFQ impulse. Indeed, in CMOS fusion computing systems, we used continuous voltages to represent the input and output logic values, instead of a single voltage pulse, without using any timing control circuits [5,6]. The same strategy could be used in SFQ circuits if we utilize multiple-flux-quantum switching properties in Josephson junctions.

4. Conclusion

A fusion computing system using SFQ circuits was developed. It consisted of an array of SFQ fusion gates (unit cells), where each unit accepted two (A and B) inputs, and generated two outputs ($A\overline{B}$ and not $\overline{A}B$). We showed that fundamental logic gates were able to be constructed by a few number of SFQ fusion gates. The operations of the unit SFQ gate and basic logic gates build by the SFQ gates were confirmed by computer simulation. With standard process parameters, we showed that the fusion computing systems could operate at 10 GHz clock, about ten times faster than conventional CMOS logic systems, if the number of fanout paths were $O(1)$.

References

[1] F. Fredkin, T. Toffoli, Int. J. Theor. Phys. 21 (1982) 219.
[2] A. Adamatzky, IEICE Trans. Electron. E87-C (2004) 1748.
[3] A. Adamatzky, Collision-Based Computing, Springer-Verlag, London, 2002, p. 411.
[4] A. Adamatzky, B. De Lacy Costello, T. Asai, Reaction–Diffusion Computers, Elsevier, London, 2005.
[5] K. Yamada, I.N. Motoike, T. Asai, Y. Amemiya, IEICE Electron. Express 3 (2006) 292.
[6] K. Yamada, T. Asai, T. Hirose, Y. Amemiya, Int. J. Unconv. Comput. 4 (2008) 45.