Supercond. Sci. Technol. **8** (1995) A46–A49. Printed in the UK

A basic circuit for asynchronous superconductive logic using RSFQ gates

I Kurosawa†, H Nakagawa†, M Aoyagi†, M Maezawa†, Y Kameda‡ and T Nanya‡

† Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan
‡ Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152, Japan

Received 18 October 1995

Abstract. This paper presents a basic circuit concept for pulse-driven asynchronous circuits using superconducting rapid single-flux-quantum (RSFQ) logic gates. Today's computer systems perform using a synchronous clock distributed over the whole system. The distribution of the clock signal becomes more difficult as the clock frequency increases because the wavelength of the clock signal becomes shorter than the system size. One way to solve this problem is to use asynchronous circuits in which all circuits perform their functions only under the law of cause and effect. We propose an implementation of the basic asynchronous circuit performing OR/AND logic operations between SFQ pulses. The operations are confirmed by SPICE simulation. We will show a design method for combinational circuits using the proposed basic circuits.

1. Introduction

Almost all computer systems of today perform using a synchronous clock signal distributed over the whole system. When the speed of logic devices increases, how to distribute a faster clock over the whole system becomes a serious problem, because the wavelength of the clock signal becomes shorter than the size of the whole system. One way to solve this problem is to use asynchronous circuits in which all circuits perform their functions only under the law of cause and effect [1, 2]. Asynchronous circuits have a very attractive potential for high-speed circuits since they can fully utilize the device speed. This paper presents a basic circuit concept for pulse-driven asynchronous circuits using superconducting rapid single-flux-quantum (RSFQ) logic gates [3]. RSFQ logic gates have advantages of high speed and low power but they function with a timing clock. The effects of this asynchronous system will become apparent.

In this paper we propose an implementation of basic asynchronous circuits performing OR/AND logic operations between SFQ pulses. The circuit operations are confirmed by SPICE simulation. Then we show a design method of combinational circuits using the proposed basic circuits.

2. Pulse-driven asynchronous logic

Logic using pulses as the information representation is unique in comparison with semiconductor logic and even with Josephson latching logic. Conventional logic is called level logic because '1' and '0' are distinguished by the signal levels. In the case of pulse logic, '1' is the existence and '0' is the absence of the pulse which behaves like a particle and runs along a signal line. We cannot represent one bit with just one signal line because we cannot know whether the data is '0' gone or '1' coming when the pulse is absent. We need another signal line for obtaining timing information of the signals. In the case of previous pulse logics using SFQ pulses, the information is represented by a pair of a data signal line and a clock signal line [3, 4]. Another way to represent the information by pulses is a dual-rail representation. $(0, 1)$ means a data '0' and $(1, 0)$ means a data '1'. $(0, 0)$ is a spacer. The dual-rail logic utilizing the dual-rail representation has some advantages. It is easy to construct an asynchronous logic system using the dual-rail logic because the timing information can be obtained by receiving the dual-rail signals themselves. NOT is easily realized by twisting the dual-rail signal lines. Figure 1 is a general expression of the dual-rail

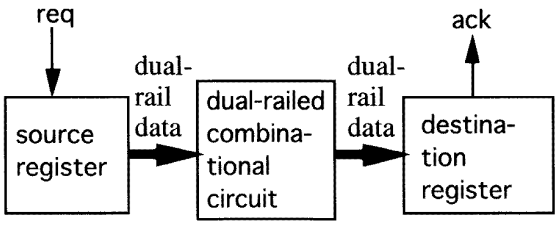

Figure 1. Dual-rail asynchronous logic.

0953-2048/95/SA0046+04$19.50 © 1995 IOP Publishing Ltd

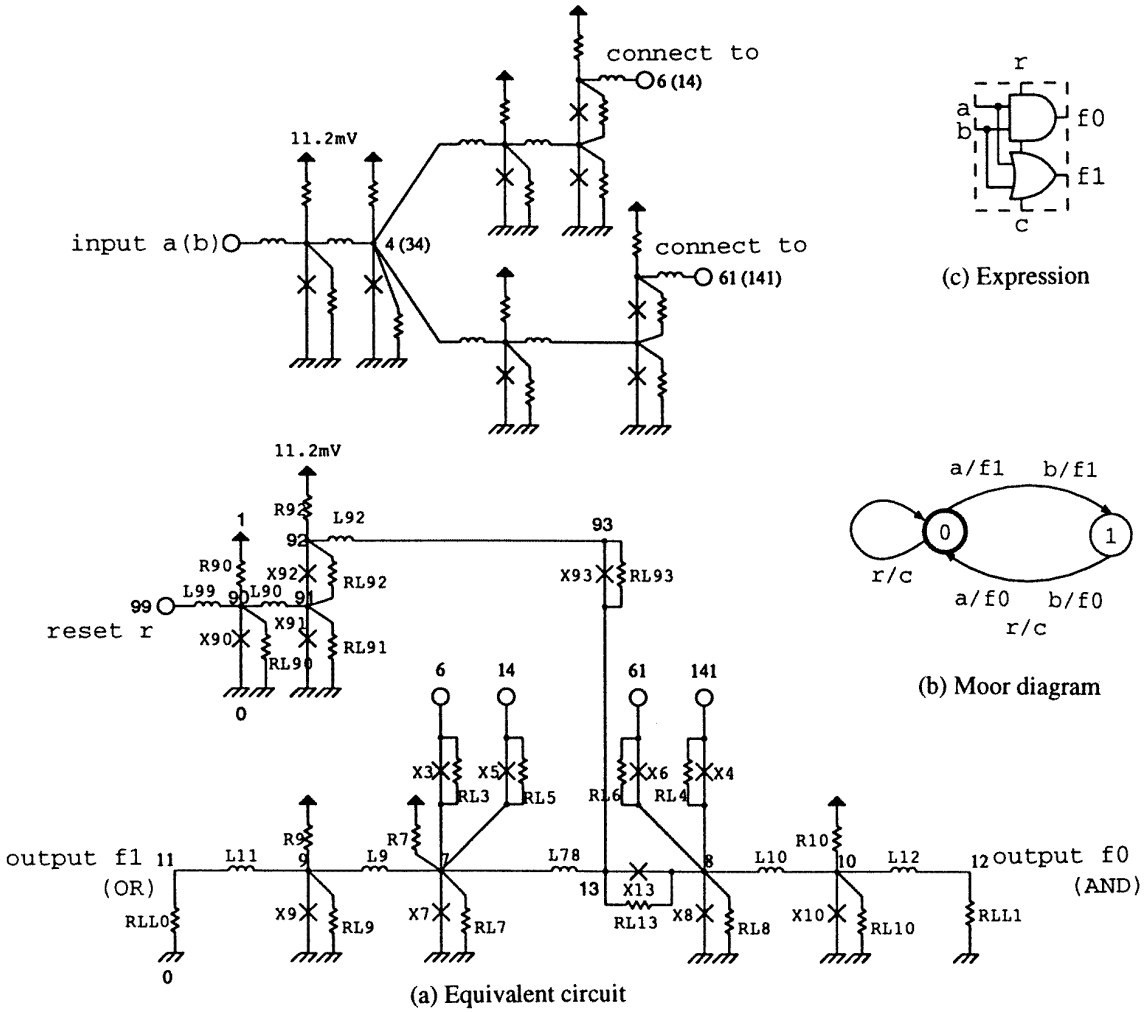

Figure 2. The proposed pulse-storage circuit functioning as OR/AND. I_0 of junctions X7, X8 and X91 is 0.28 mA. I_0 of X13 is 0.24 mA. Other junctions have 0.2 mA I_0. R92 is 53 Ω, R7 is 70 Ω and other feed resistors are 75 Ω. L92 is 0.1 pH, L78 is 13 pH and other inductances are 5 pH.

asynchronous logic. In general, all processing of data, e.g. in a computer, is represented by combinational logics from a source register to a destination register. This process starts at a request signal (req) to the source register. The dual-rail data from the source register are processed in the dual-railed combinational circuits and the result is sent to the destination register. When the destination register receives the data, an acknowledge signal (ack) is returned. The ack signal is used for resetting all the combinational circuits and requesting the next data to be processed.

3. A basic circuit: pulse-storage circuit functioning as OR/AND

All the logics are constructed with OR, AND and NOT. NOT is realized by twisting the signal rails in the asynchronous pulse logic. Therefore, an OR/AND circuit is needed as a basic circuit. We propose such a basic circuit for dual-rail asynchronous pulse logic based on the recently developed RSFQ circuit family [3]. Figure 2 shows the basic circuit functioning as OR/AND constructed with some basic RSFQ gates. A symbol X means a Josephson junction and a parallel resistor is a shunt resistance for obtaining a non-hysteretic I–V characteristic of the junction. In general, it is not guaranteed that the arrivals of two input pulses are coincident. Therefore, a storage circuit for the first arrival pulse is needed for two input AND operation. The basic circuit, called the pulse-storage circuit, is a modified circuit of the T flip-flop of the RSFQ logic family combined with the pulse splitter and the buffer of the same family of circuits. The pulse-storage circuit has three inputs (a and b, input signals; r, reset signal) and three outputs (f0 and f1, output signals; c, reset completion signal). These are different from the original T flip-flop circuit. We have one pulse from f0 output when two input pulses have arrived. On the other hand, we have one pulse from f1 output when at least one input pulse has arrived. Therefore, f0 and f1 correspond to AND and OR, respectively. The expression of the circuit is shown in figure 2(*c*). Unfortunately the reset completion signal c

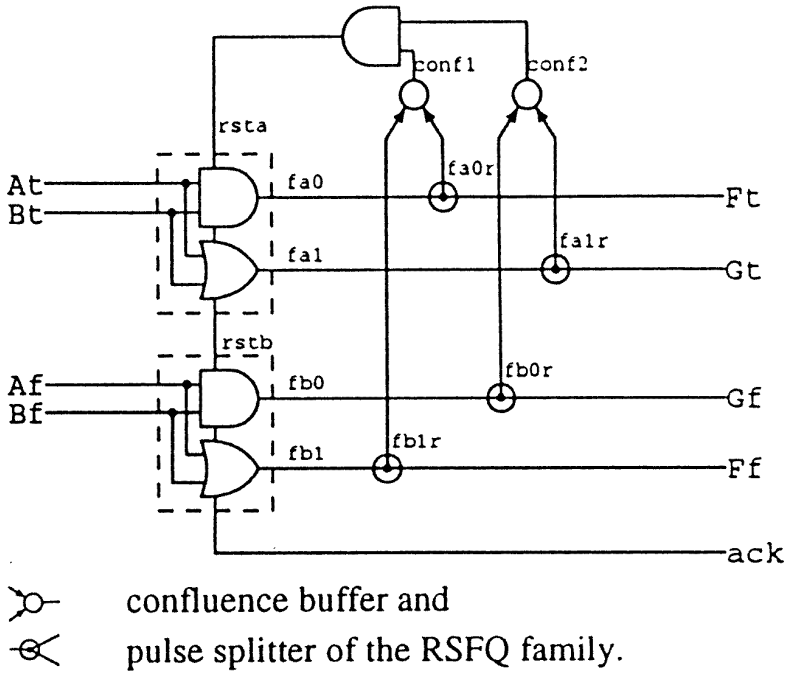

Figure 4. Dual-railed AND/OR circuits using the pulse-storage circuits.

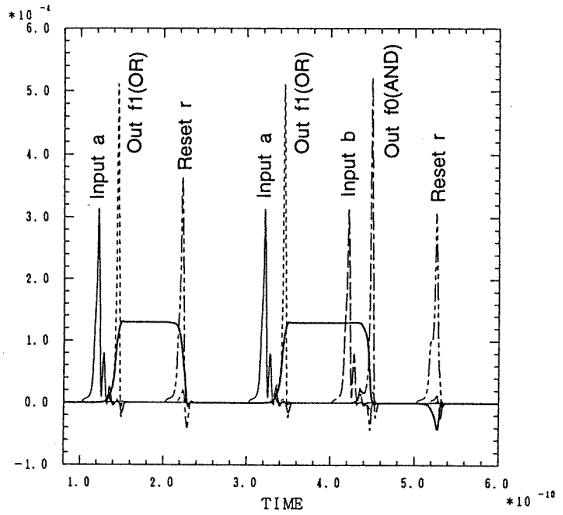

Figure 3. Simulated results of the pulse-storage circuit. A thick solid line corresponds to the current of the SQUID loop.

cannot be easily obtained from this circuit. Therefore, splitting the r signal is used for the c signal.

The functions of the pulse-storage circuit are simulated and the proper operations are confirmed by using the FSPICE program. One of the simulated waveforms is shown in figure 3. When an SFQ pulse arrives at the input a, it induces an SFQ trap in the SQUID loop X7, L78, X13, X8 (which corresponds to the state transition $0 \rightarrow 1$), and makes an output pulse from f1 (OR). The next SFQ pulse arrival at the input b clears the trap of the SQUID loop (state $1 \rightarrow$ state 0) and simultaneously makes an output pulse from f0 (AND). If there is no second pulse, the arrival of a reset pulse at the input r triggers the reset of the SQUID loop (state $1 \rightarrow$ state 0) by switching the Josephson junction X13. At that time no output signal is made from f0 or f1. When the state is 0, this reset pulse does not affect the state of the SQUID loop. Therefore, the reset pulse guarantees the state 0 of the circuit.

It is important that the pulse-storage circuit operates properly even when pulses arrive at inputs a and b simultaneously. The computer simulation shows that the simultaneous inputs make both f1 and f0 output one pulse each without any change of the state 0. Consequently, this circuit guarantees its timing-free operation. The operating margin ranges from -7% to $+9\%$ of the nominal 11.2 mV supply voltage. The delay times at the nominal supply voltage are 25 ps, 30 ps and 10 ps for OR, AND and reset operation, respectively. The design uses Josephson junctions made from $Nb/AlO_x/Nb$ with 2 kA cm^{-2} critical current density. The circuit parameters are not optimized, so the margin is rather narrow.

An example of how to use the pulse-storage circuit in the dual-rail synchronous logic is shown in figure 4. Its functions are both OR and AND of the inputs A(At, Af) and B(Bt, Bf) in the dual-rail representation. Outputs F(Ft, Ff) are the dual-rail AND output of A B. G(Gt, Gf) corresponds to OR of A+B. The timing information is obtained through a simple AND function whose inputs are incoming of the outputs of two pulse-storage circuits. The obtained timing signal resets the pulse-storage circuits and is split to make an ack signal.

4. Summary

Problems concerning the clock distribution become more serious in higher-speed systems using ultra-high-speed circuits such as RSFQ. Therefore, some asynchronous methods have been proposed and studied for timing of high-speed RSFQ circuits. However, they need a line for a local clock signal in addition to the data signal lines. Even though they do not need a global clock signal, the locally clocked circuits are also designed on complete knowledge about the delay and timing of the circuits. As the circuit complexity increases, it becomes more difficult to know all about the timing information of the circuits before their implementation and operation. By introducing a dual-rail logic the timing information can be easily obtained from the data themselves. Therefore, real asynchronous circuits can be designed without knowledge of the delay and timing.

A circuit concept for pulse-driven asynchronous logics has been proposed on the base of dual-rail logic. An implementation of the basic asynchronous circuit, the pulse-storage circuit, performing OR/AND logic operations between SFQ pulses, has been constructed using RSFQ logic gates. The circuit operations are confirmed by SPICE simulation. Using the proposed pulse-storage circuit all of the dual-railed combinational logics can be constructed.

Acknowledgments

The authors (IK, HN, MA and MM) would like to thank Drs T Sakamoto, S Takada and A Shoji for their continuous support and encouragement. They also thank Dr Stein R Gjoen for his critical reading of this paper.

References

[1] Nanya T 1990 *IEICE Technical Report* FTS90–45 (in Japanese)
[2] Nanya T, Ueno Y, Kagotani H, Kuwako M and Takamura A 1994 *IEEE Design Test* **11** 50
[3] Likharev K K and Semenov V K 1991 *IEEE Trans. Appl. Supercond.* **AS-1** 3
[4] Nakajima K, Mizusawa H, Sugahara H and Sawada Y 1991 *IEEE Trans. Appl. Supercond.* **AS-1** 29