# Static Timing Analysis of Rapid Single-Flux-Quantum Circuits

Takahiro Kawaguchi Kazuyoshi Takagi Naofumi Takagi
Department of Communications and Computer Engineering, Kyoto University
Kyoto, Japan
kawaguti@lab3.kuis.kyoto-u.ac.jp

**Abstract— We propose a method for calculating pulse arrival timing at all gates in an RSFQ circuit and a new definition of timing slacks of gates. In the proposed method, the total path delay, the total length of PTLs and the number of PTL transmitters/receivers on a path are also calculated.**

## I. Introduction

Superconducting rapid-single-flux-quantum (RSFQ) circuit technology is an emerging technology with high switching speed and low energy consumption[1]. An RSFQ logic gate consists of superconducting loops with Josephson junctions and inductances. Fabrication of an RSFQ circuit with more than 10,000 Josephson junctions has become possible [2, 3, 4]. An RSFQ logic circuit uses a voltage pulse and a magnetic flux quantum as a carrier of information and for state representation, respectively. Cell-based design is adopted in designing RSFQ circuits. A set of logic gates for RSFQ circuits is predefined as a cell library [5]. Behavior of each gate which is represented by pulse transferring and state transition is defined in the cell library.

To design large scale RSFQ circuits, computer-aided design (CAD) tools are indispensable. Because RSFQ circuits use voltage pulses and behave differently from CMOS circuits, CAD tools for CMOS circuits can not be used in RSFQ circuits as they are without change. Tools for clock tree synthesis [6], placement and wire routing[7, 8], circuit description [9, 10], static timing analysis[11] and formal verification [12] are developed for RSFQ circuit designs.

RSFQ circuits are driven by voltage pulses. Representation of logic values using voltage pulse is different from that using voltage level. In general, the logic values "1" and "0" are represented by the presence and the absence of a pulse, respectively. Such representation needs to distinguish the logic value "0" from the state of "no signal". In most common logic representation using a synchronizing clock, a time frame for a gate is defined as a time section partitioned by clock pulses and the absence of a pulse in the time frame represents logic value "0" at the time frame. Therefore, a gate with clock supply, which is called a clocked gate, is commonly used in RSFQ circuits. Because the clocked gates store the data, RSFQ circuits is often designed with gate-level pipelines.

Switching of RSFQ circuits is faster than that of CMOS circuits. To achieve multi-giga-hertz circuit, skewed clocking scheme is often chosen in RSFQ circuits. There exist several clocking schemes with different behavior of a gate. In the most common clocking scheme in RSFQ circuits, called concurrent-flow clocking, a data pulse arrives after the corresponding clock pulse and the gate behaves as a logic gate combined with a delay flip-flop. In the clocking scheme called clock-follow-data clocking, a data pulse arrives before the corresponding clock pulse and the gate behaves as a logic gate in a combinational circuit.

In RSFQ circuits, the order of pulse arrival times of inputs of a gate effects the behavior of the gate. A simple example of the effect of the order is a clocked gate in the concurrent-flow clocking and the clock-follow-data clocking. A clocked gate in the concurrent-flow clocking has the different order of pulse arrival times of inputs from that in the clock-follow-data clocking even if these gates are the same type, the behavior of a gate are different from that of the other gate.

In this paper, we propose a timing analysis method to determine the order of pulse arrival times at each gate in RSFQ circuits. From the information provided as a logic cell library, we calculate the path delay time, timing slacks and the minimum clock period. Based on our method, temporal behavior of the circuit is determined and functional verification becomes possible.

## II. RSFQ gates and circuits

An RSFQ logic gate consists of some superconducting loops with Josephson junctions (JJs) and inductances. When a voltage pulse arrives at a loop, a flux quantum, i.e. an RSFQ, can be trapped or released. When an RSFQ is released, a short voltage pulse is produced. A voltage pulse and an RSFQ are used as a carrier of information and for state representation, respectively.

Cell-based design is adopted in designing RSFQ circuits [5]. Timing parameters, i.e. delay and timing constraints, and behavior of each gate are defined in a cell library for RSFQ circuits. While an RSFQ circuit is working, bias current is supplied to all gates in the circuit. The timing parameters depend on the amount of

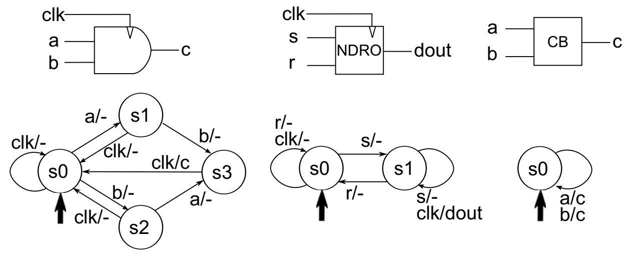

Fig. 1. State transition diagrams of a clocked AND, an NDRO and a CB

bias current to a gate. A gate with clock supply, which is called a clocked gate, is commonly used in RSFQ circuits. There are clocked AND, OR, EXOR, NOT, etc. A delay flip-flop (DFF) is also a clocked gate. Some gates do not have clock supply. As interconnections in RSFQ circuits, Josephson-transmission-line (JTL) and Passive-transmission-line(PTL) cells are used. JTL cells are used for short interconnections and delay elements to adjust timing. PTL cells are used for long interconnections. To connect PTL cells to the other cells, PTL transmitter and receiver cells are arranged on the source and the destination of a row of PTL cells, respectively.

Clocking and timing designs are important factors affecting the performance of an RSFQ circuit because gates switch with high speed and wiring delay is not negligible. Zero skew clocking, which is commonly used in CMOS circuits, is not used in most RSFQ circuits. To achieve multi-gigahertz circuit, flow-clocking is often chosen [13]. In flow-clocking, a skewed clock pulse is distributed to clocked gates along data path and an interval between a clock pulse and a data pulse is shorter than that in zero skew clocking. Several flow-clocking schemes are used in RSFQ circuits. One scheme is called concurrent-flow clocking. In the concurrent-flow clocking scheme, a data pulse arrives after the corresponding to clock pulse and a gate behaves like as a logic gate combined with a DFF. There is another flow-clocking scheme, called clock-follow-data clocking. In the clock-follow-data clocking scheme, a data pulse arrives before the corresponding to clock pulse and a gate behaves like as a gate in a combinational circuit.

There often exists an RSFQ circuit using logic gates without clock supply. Typical examples of such gates are Concluence Buffer (CB) and Non-Destructive Read Out (NDRO). A CB has no clock input and is used as an asynchronous OR gate. Although an NDRO has a clock input *clk*, the clock input sometimes connects to a data signal line rather than a clock signal line [3, 4, 14].

The digital behavior of gates in RSFQ cell libraries is described by using a hardware description language. Timing constraints between an ordered pair of inputs, delay from an input to an output, state transitions by a pulse arrival at an input and a pulse generation to an output by a pulse arrival at an input on a state are described as behavior of an RSFQ gate. The descriptions of RSFQ gates enable logic simulation of RSFQ circuits. FIg. 1. shows state transition diagrams of RSFQ gates.

## III. Static timing analysis of RSFQ circuits

### A. Overview

Temporal behavior of RSFQ circuits has troublesome aspects because RSFQ circuits work by pulse logic and most logic gates are clocked. Firstly, clock cycles are defined locally for each gate and no global synchronous clock can be defined. Second, depending on the gate types, timing of the outputs of a gate can depend on the timing of the data inputs or the clock input. Third, timing dependences and restrictions can exist also between data inputs, and there can be maximum restriction for the input timing.

In this section, we describe a method to calculate pulse arrival timing at all gates in an RSFQ circuit. We assume that no timing loop exists in given circuit, that is, timing of a gate input does not depend on itself. In the proposed method, the total path delay, the total length of PTLs and the number of PTL transmitters/receivers on a path are also calculated. These values are required because the bias current dependences of PTL wire delay and the transmitter/receiver delay are different from that of active logic gates and JTL.

In static timing analysis, compared to the conventional simulation-based analysis, no specific input patterns are needed and the timing analysis and verification results do not depend on the patterns.

Calculation of the path delay of an RSFQ circuit is slightly different from that of a CMOS circuit. In an RSFQ circuit, memory elements such as delay flip-flop are not distinguished from other logic gates, because all logic gates have memory functions. Timing parameters such as delay and timing constraint of a gate are affected by bias current, timing jitter and fabrication variability.

In general, timing constraints of an RSFQ gate are given as setup and hold time constraints [15]. The setup and hold time constraints represent the relationship between a data input and a clock input of a gate. The concurrent-flow clocking and the clock-follow-data clocking have different definitions of setup and hold constraints. If an RSFQ circuit employs plural clocking schemes with the different definitions, the clocking schemes need to be declared on each gate to use setup and hold constraints. In addition, timing constraints of an RSFQ gate may also exist between two data inputs. For example, data inputs *s* and *r* of an NDRO gate have an interval time constraint. If pulses on the inputs arrive closely within the minimum interval time defined in the constraint, the gate can perform incorrectly. On the other hand, inputs *a* and *b* of a CB gate have another interval time constraint. When a pulse on each of the inputs arrives closely within the maximum interval time defined in the constraint, only one pulse appears on an output $c$ of the CB. When these two pulses are farther than the maximum interval time defined in the constraint, two pulses appear on the output $c$. Because the presence of two pulses on a line in a time frame is not assumed, a large interval time between $a$ and $b$ of CB can lead to a timing violation.

## B. *Calculation of delay*

In this paper, we refer to an input pin of a gate as an input of a gate. Similarly, an output pin of a gate is referred to as an output of a gate.

Timing dependency between an input and an output of each gate is described in the gate library. When a pulse on an output of a gate can be produced by a pulse arrival on an input of the gate, pulse arrival time of the output depends on that of the input. A pulse arrival time of an output often depends on those of several inputs.

The earliest pulse arrival time to output $o$, $T_e(o)$, and the latest pulse arrival time to $o$, $T_l(o)$, are calculated by the followings,

$$T_l(o) = \max_{i \in ID(o)} (T_l(i) + D(i,o)), \quad (1)$$
$$T_e(o) = \min_{i \in ID(o)} (T_e(i) + D(i,o)), \quad (2)$$

where the set of inputs whose pulse arrival times affect that of $o$ is $ID(o)$ and the delay from $i$ to $o$ is $D(i,o)$. The pulse arrival times to input $i$, $T_e(i)$ and $T_l(i)$ are calculated by the followings,

$$T_l(i) = (T_l(fi) + D(fi,i)), \quad (3)$$
$$T_e(i) = (T_e(fi) + D(fi,i)), \quad (4)$$

where the fanin of $i$ is $fi$ which is an output of a gate.

Algorithm 1 shows the calculation process of the path delay in an RSFQ circuit. This algorithm calculate pulse arrival time of outputs of a gate by Eqs. 1 and 2. At the same time, the number of PTL transmitters and PTL length, which equals to the number of PTL cells, are calculated. PTL receiver must have the corresponding PTL transmitter. Therefore, the number of PTL receiver is not calculated in this algorithm. This algorithm assumes that the arrival time of an output of a gate in the circuit does not depend on itself. If there exists such output, pulse arrival time of it cannot be defined.

## C. *Calculation of timing slack and clock period*

We classify these interval time constraints of a gate into two types. They are the minimum and the maximum interval time constraints. Each ordered pair of inputs of a gate has one of these constraints or does not have an interval time constraint. We consider an ordered pair of inputs $(x, y)$ such that a pulse of $y$ arrives after a pulse arrival at $x$. When $(x, y)$ has the minimum interval time constraint, a pulse on $y$ should arrive after the elapse of *the minimum interval time* from a pulse arrival time of $x$ at the earliest. When $(x, y)$ has the maximum interval time constraint, a pulse on $y$ should arrive by the elapse of *the maximum interval time* from a pulse arrival time of $x$ at the latest. The interval time constraints do not depend on clocking schemes. Therefore, the declaration of a clocking scheme on each gate is not necessary in the proposed verification method.

```
Input: an RSFQ circuit
Output: pulse arrival times of primary outputs and
        inputs of all gates
// $FO(o)$ is the set of inputs which are
   fan-outs of output $o$
// $ID(o)$ is the set of inputs whose pulse
   arrival times affect that of output $o$
// $OD(i)$ is the set of outputs whose pulse
   arrival times depend on that of input $i$
$I \leftarrow$ the set of inputs connecting to primary inputs;
while $I \neq \emptyset$ do
    Pick an input $i \in I$;
    $I \leftarrow I - \{i\}$;
    foreach $od \in OD(i)$ do
        if $\forall id \in ID(od), T_e(id)$ and $T_l(id)$ are
        calculated then
            Calculate $T_e(od)$ and $T_l(od)$ by Eq. 1 and
            Eq. 2;
            if od is an output of a PTL transmitter
            then
                $the\_number\_of\_PTL\_transmitter++$
            end
            if od is an output of a PTL cell then
                $the\_number\_of\_PTL\_cell++$
            end
            foreach $fo \in FO(od)$ do
                Calculate $T_e(fo)$ and $T_l(fo)$ by Eq. 3
                and Eq. 4;
            end
            $I \leftarrow I \cup FO(od)$;
        end
    end
end
```
**Algorithm 1:** Calculation of delay

For an ordered pair of inputs $(x, y)$ of gate $g$, the timing slack $TS(g, x, y)$ is defined in three cases: (a) $T_l(y)$ is larger than $T_e(x)$ and $(x, y)$ has the minimum interval time constraint (b) $T_l(y)$ is larger than $T_e(x)$ and $(x, y)$ has the maximum interval time constraint (c) otherwise, namely, $T_l(y)$ is not larger than $T_e(x)$ or $(x, y)$ does not have the minimum or the maximum interval time constraint. Fig. 2(a) shows the timing slack calculated by the minimum interval time constraint. Fig. 2(b) shows the timing slack calculated by the maximum interval time constraint. $TS(g,x,y)$ is calculated by the following,

$$TS(g,x,y) = \begin{cases} T_e(y) - T_l(x) - IT_{min}(x,y) & \text{(a)} \\ T_e(x) + IT_{max}(x,y) - T_l(y) & \text{(b)} \\ \text{no value} & \text{(c)} \end{cases} \quad (5)$$

where the minimum and the maximum interval times of $(x,y)$ are $IT_{min}(x,y)$ and $IT_{max}(x,y)$, respectively. If $TS(g,x,y)$ or $TS(g,y,x)$ is negative, a timing violation may occur between $x$ and $y$.

A clocked AND gate has the minimum interval time constraints on $(a, clk)$, $(clk, a)$, $(b, clk)$, $(clk, b)$, $(a,a)$, $(b,b)$ and $(clk, clk)$. This gate does not have the interval time constraints on $(a,b)$ and $(b,a)$. A CB gate has the minimum interval time constraints on $(a,a)$ and $(b,b)$ and the maximum interval time constraints on $(a,b)$ and $(b,a)$.

To guarantee satisfying the interval time constraints between inputs in a time frame and the next time frame, we calculate the minimum clock period. Fig. 2 shows the minimum clock period of each gate. The minimum clock period of an ordered pair of inputs $(x,y)$ of gate $g$, $CP_{min}(g,x,y)$, and the minimum clock period of gate $g$, $CP_{min}(g)$, are calculated by the followings,

$$CP_{min}(g,x,y) = \{T_l(y) - T_e(x) + IT(y,x)\}, \quad (6)$$
$$CP_{min}(g) = \max_{x,y \in I(g)} (CP(g,x,y)), \quad (7)$$

where $I(g)$ is the set of inputs of gate $g$ and $IT(y,x)$ is $IT_{min}(y,x)$ in Fig. 2(a), $IT_{max}(y,x)$ in Fig. 2(b) or 0 in Fig. 2(c). When an interval time constraint does not exist on $(y,x)$, $IT(y,x)$ is 0. $IT(y,x)$ includes $IT_{max}(y,x)$, because $IT_{max}(y,x)$ is the interval time constraint in a time frame. If the clock period is smaller than $T_l(y) - T_e(x) + IT_{max}(y,x)$, a pulse on $x$ in a time frame is dealt with as a pulse in the previous frame. The minimum clock period of the circuit is

$$CP_{min} = \max_{\forall g}(CP_{min}(g)). \quad (8)$$

Fig. 2 shows examples of calculated timing slacks and clock period. Calculated timing slacks and clock period of a clocked AND gate $g1$, which has inputs $a, b$ and $clk$, is shown in Fig. 2 (a). In this figure, $TS(g1, b, clk)$, is calculated as $T_e(b) - T_l(clk) - IT_{min}(b, clk)$ because $T_l(clk)$ is larger than $T_e(b)$ and $(b, clk)$ has the minimum interval time constraint. Similarly, $TS(g1, a, clk)$ has a calculated value. $TS(g1, a, b)$ and $TS(g1, b, a)$ have no value because there are no timing constraints on $(a,b)$ and $(b,a)$. $TS(g1, clk, a)$ and $TS(g1, clk, b)$ have no value because the latest arrival time of the successor of an ordered pair is not larger than the earliest arrival time of the predecessor of the ordered pair. The minimum clock period defined by $(a, clk)$, $CP_{min}(g1, a, clk)$ is shown in Fig. 2 (a). If $CP_{min}(g1, a, clk)$ is larger than any calculated clock period on $g1$, $CP_{min}(g1, a, clk)$ is the minimum clock period

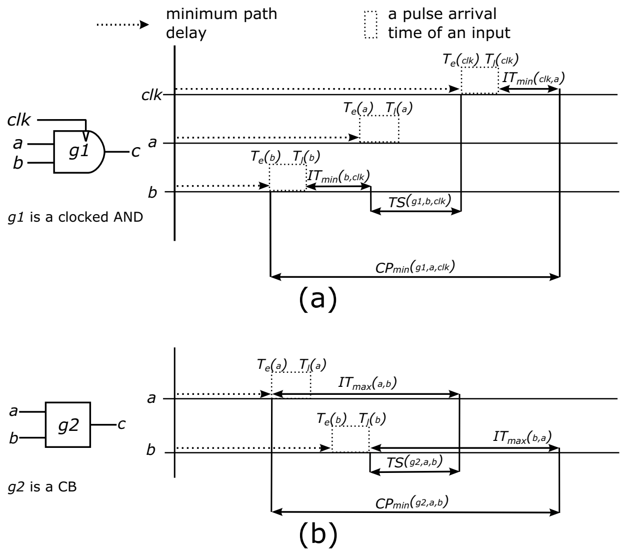

Fig. 2. Calculated timing slack and clock period
(a) $TS(g1, a, b) =$ no value,
$TS(g1, b, a) =$ no value,
$TS(g1, a, clk) = T_e(clk) - T_l(a) - IT_{min}(a, clk)$,
$TS(g1, clk, a) =$ no value,
$TS(g1, b, clk) = T_e(b) - T_l(clk) - IT_{min}(b, clk)$,
$TS(g1, clk, b) =$ no value,
and $CP_{min}(g1, a, clk) = T_l(clk) - T_e(a) + IT_{min}(clk, a)$
(b) $TS(g2, a, b) = T_e(a) + IT_{max}(a, b) - T_l(b)$,
$TS(g2, b, a) =$ no value
$CP_{min}(g2, a, b) = T_l(b) - T_e(a) + IT_{max}(b, a)$

defined by $g1$. Fig. 2 (b) shows calculated timing slacks and clock period of CB gate $g2$ which has the maximum interval time constraints.

## IV. Conclusion

We have proposed a method for static timing analysis of RSFQ circuits. To verify the timing of RSFQ circuits, we have introduced a new definition of timing slacks of RSFQ gates. Additionally, analysing the total length of PTLs is useful to design a circuit which has large bias current margin.

## Acknowledgments

This work was supported in part by CREST JST, ALCA JST and VLSI Design and Education Center (VDEC), the University of Tokyo in Collaboration with Cadence Design Systems, Inc.

## References

[1] Konstantin K Likharev and Vasilii K Semenov. RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock-

frequency digital systems. *Applied Superconductivity, IEEE Transactions on*, 1(1):3–28, 1991.

[2] Y Yamanashi, M Tanaka, A Akimoto, H Park, Y Kamiya, N Irie, N Yoshikawa, A Fujimaki, H Terai, and Y Hashimoto. Design and implementation of a pipelined bit-serial SFQ microprocessor, core1$\beta$. *IEEE Transactions on Applied Superconductivity*, 17(2):474–477, 2007.

[3] Heejoung Park, Yuki Yamanashi, Kazuhiro Taketomi, Nobuyuki Yoshikawa, Masamitsu Tanaka, Koji Obata, Yuki Ito, Akira Fujimaki, Naofumi Takagi, Kazuyoshi Takagi, et al. Design and implementation and on-chip high-speed test of SFQ half-precision floating-point adders. *IEEE Transactions on Applied Superconductivity*, 19(3):634–639, 2009.

[4] Hiroshi Hara, Koji Obata, Heejoung Park, Yuki Yamanashi, Kazuhiro Taketomi, Nobuyuki Yoshikawa, Masamitsu Tanaka, Akira Fujimaki, N Takagi, Kazuyoshi Takagi, et al. Design, implementation and on-chip high-speed test of SFQ half-precision floating-point multiplier. *IEEE Transactions on Applied Superconductivity*, 19(3):657–660, 2009.

[5] S Yorozu, Y Kameda, H Terai, A Fujimaki, T Yamada, and S Tahara. A single flux quantum standard logic cell library. *Physica C: Superconductivity*, 378:1471–1474, 2002.

[6] Kazuyoshi Takagi, Shota Takeshima, Masamitsu Tanaka, and Naofumi Takagi. Layout-driven skewed clock tree synthesis for superconducting SFQ circuits. *IEICE transactions on Electronics*, E94-C(3):288–295, 2011.

[7] Yoshio Kameda and Shinichi Yorozu. Automatic Josephson-transmission-line routing for single-flux-quantum cell-based logic circuits. *IEEE Transactions on Applied Superconductivity*, 13(2):519–522, 2003.

[8] Masamitsu Tanaka, Koji Obata, Shota Takeshima, Kazuyoshi Takagi, Naofumi Takagi, Hiroyuki Akaike, and Akira Fujimaki. Automated passive-transmission-line routing tool for single-flux-quantum circuits based on A* algorithm. *IEICE transactions on electronics*, 93(4):435–439, 2010.

[9] F Matsuzaki, N Yoshikawa, M Tanaka, A Fujimaki, and Y Takai. A behavioral-level HDL description of SFQ logic circuits for quantitative performance analysis of large-scale SFQ digital systems. *Physica C: Superconductivity*, 392:1495–1500, 2003.

[10] Kazuyoshi Takagi, Nobutaka Kito, and Naofumi Takagi. Circuit description and design flow of superconducting SFQ logic circuits. *IEICE Transactions on Electronics*, E97-C(3):149–156, 2014.

[11] Yoshio Kameda, Shinichi Yorozu, Yoshihito Hashimoto, Hirotaka Terai, Akira Fujimaki, and Nobuyuki Yoshikawa. High-speed demonstration of single-flux-quantum cross-bar switch up to 50 GHz. *IEEE Transactions on Applied Superconductivity*, 15(1):6–10, 2005.

[12] Kazuyoshi Takagi, Motonori Sato, Masamitsu Tanaka, and Naofumi Takagi. A verification method of pipeline processing behavior of superconducting singl-flux-quantum pulse logic circuits. In *16th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI2010)*, R2-17, pages 208–231, 2011.

[13] Oleg A Mukhanov, Paul D Bradley, Steven B Kaplan, Sergey V Rylov, and AF Kirichenko. Design and operation of RSFQ circuits for digital signal processing. In *Proc. 5th Int. Supercond. Electron. Conf*, pages 27–30, 1995.

[14] Masamitsu Tanaka, Hiroyuki Akaike, Akira Fujimaki, Yuki Yamanashi, Nobuyuki Yoshikawa, Shuichi Nagasawa, Kazuyoshi Takagi, and Naofumi Takagi. 100-GHz single-flux-quantum bit-serial adder based on 10-niobium process. *IEEE Transactions on Applied Superconductivity*, 21(3):792–796, 2011.

[15] Kris Gaj, Eby G Friedman, and Marc J Feldman. Timing of multi-gigahertz rapid single flux quantum digital circuits. In *High Performance Clock Distribution Networks*, pages 135–164. Springer, 1997.