A Clock Distribution Scheme for Large RSFQ Circuits

Krzysztof Gaj, Eby G. Friedman, and Marc J. Feldman
Department of Electrical Engineering, University of Rochester, Rochester, NY 14627

Andrzej Krasniewski
Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract—A primary issue in maximizing the performance of large scale synchronous digital systems is the clock distribution scheme. We present a novel clocking scheme, developed specifically for RSFQ logic, which is based on the concurrent flow of the clock and data signals. The scheme permits the circuit throughput to be independent of inter-cell connection delays and significantly reduces the dependence of the throughput on the clock-to-output delay of the cells. Concurrent flow clocking is particularly well suited for structured architectures. The simulated maximum clock frequency of an RSFQ decimation digital filter currently under development at the University of Rochester can be as much as seven times higher using concurrent-flow clocking rather than conventional (counterflow) clocking. This advantage, however, is reduced to a factor of two due to fabrication process parameter variations in present day superconductive technologies.

I. INTRODUCTION

Full exploitation of the speed of Rapid Single Flux Quantum (RSFQ) logic will require the proper choice of system level architecture. An essential issue in system level design is the choice of synchronization strategy, ranging from fully synchronous to fully asynchronous architectures. RSFQ is the first superconducting digital technology that can use asynchronous schemes as well as synchronous schemes and the possibility of constructing superfast large asynchronous processors has therefore attracted significant interest [1]-[4].

The optimal choice of RSFQ logic architecture should consider the experiences of high-speed semiconductor devices. Although it is often asserted that asynchronous circuits have a performance advantage, a detailed analysis of semiconductor-based logic has shown that asynchronous architectures hardly ever outperform fully synchronous-based architectures. In fact, fully synchronous pipelined clocking almost always outperforms asynchronous timing schemes [5].

Since pipelining is common in digital signal processors, an important near-term application of RSFQ logic [1], [4], asynchronous RSFQ schemes begin with a significant disadvantage. Furthermore, other well-known advantages of synchronous circuits exist, such as smaller area, improved testability, and well established and easier design techniques. We therefore expect that as RSFQ logic matures, it will follow the design path of its semiconductor counterparts.

The primary argument against a synchronous architecture is the deleterious effects of clock skew on circuit performance [2], [4]. At the high frequencies expected with RSFQ circuits, clock skew can become the dominant factor limiting the minimum clock period. One solution for this problem in RSFQ circuits is the use of complex clock distribution networks, such as load-balanced H-trees [4], [6] targeted to minimize clock skew. These networks have some application to semiconductor circuits but are rather inadequate for low-fanout RSFQ logic, as complex clock distribution networks require a large number of splitters integrated into a complex and area inefficient clock distribution tree.

In this paper we propose a new approach to eliminating the deleterious influence of clock skew on RSFQ circuits: *concurrent flow clocking*. We show that by distributing the clock signal in the direction of the data flow, circuit performance can be enhanced as compared with schemes that attempt to eliminate or minimize clock skew.

Similar synchronization schemes have been developed for semiconductor logic [6]–[8] but have not as yet been widely adopted. The main reasons are conservative design conventions used within industry, relatively small performance improvements (up to 40%) as compared with schemes targeted to produce zero clock skew, and difficulties in implementing well-controlled delay lines within semiconductor-based clock distribution networks.

For RSFQ logic, the concurrent flow of clock and data has been applied to simple shift register [1], [2], [9] and full-adder structures [10]. The scheme permits either fast initialization of shift registers [1], or construction of fast circular shift registers [9]. Our methodology generalizes and optimizes this approach to make non-zero skew clock distribution applicable and efficient for a variety of synchronous RSFQ circuits. We show that, due to specific features of RSFQ logic, the potential performance improvements of applying this synchronous scheme may be significantly higher than in semiconductor devices.

Our paper is organized as follows. In the following section we present our scheme and compare it with the synchronization scheme based on the counterflow of clock and data signals. In section III, we provide a detailed comparison of concurrent and counterflow clocking, considering the deviations of circuit parameters due to variations in the fabrication process. Section IV is a brief prescription for using the equations to design a clock distribution network while accounting for these parameter deviations. In section V, as an example, we apply both clocking schemes to the design of a decimation digital filter currently under development at the University of Rochester. Finally, we present our conclusions in section VI.

Manuscript received October 18, 1994; revised February 28, 1995.
Financial support for this research was provided by the University Research Initiative at the University of Rochester, sponsored by the Army Research Office under Grant No. DAAL03-92-G-0112.

II. General Idea of Counterflow and Concurrent Flow Clocking Schemes

In typical structured LSI and VLSI architectures, such as systolic arrays, the flow of data is closely associated with the circuit topology. In particular, data is most often exchanged between physically neighboring cells. We assume that the flow of the clock signal in RSFQ circuits is also closely related to the topology of the circuit and therefore to the flow of the data. This assumption permits a significant reduction in the complexity of the clock distribution network, including the number of splitters and Josephson transmission lines (JTLs) that constitute the network. Thus, two basic clocking schemes are possible:

In the *counterflow scheme*, the clock propagates in the opposite direction to that of the data flow. In the *concurrent flow scheme*, the subject of this paper, the clock flows in the same direction as the data. This is illustrated in Fig. 1. In both schemes extra delay ($\Delta_{DATA\text{-}SYN}$) may need to be added to the data path to avoid races in the circuit (i.e., the propagation of data pulses through several consecutive clocked gates within one clock cycle). However, only in the concurrent flow clocking, an additional delay ($\Delta_{CLK\text{-}SYN}$) added to the clock path can permit the designer to more efficiently utilize the intrinsic speed of the RSFQ cells. Timing diagrams describing the exchange of data between two cells are shown in Fig. 2. Necessary and sufficient conditions for correct data exchange is described for both clocking schemes, as follows (see Figs. 1 and 2 for the notation):

$$t.hold(2) \le data.position(2) \le T_{CLK} - t.setup(2). \quad (1)$$

The first inequality eliminates any race conditions in the circuit while the second inequality imposes limits on the minimum clock period.

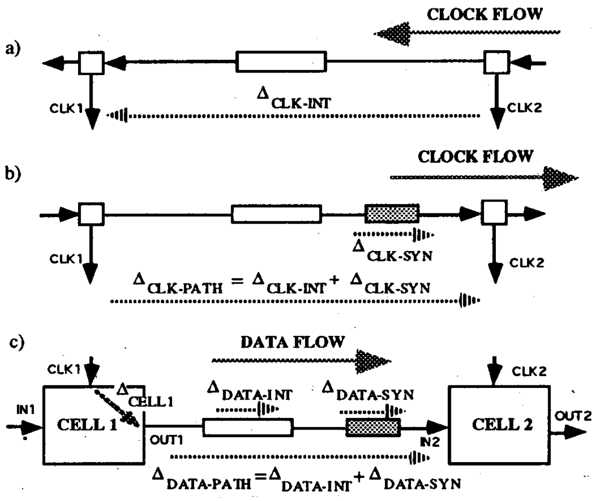
a)
CLOCK FLOW
CLK1
$\Delta_{CLK\text{-}INT}$
CLK2
b)
CLOCK FLOW
CLK1
$\Delta_{CLK\text{-}SYN}$
CLK2
$\Delta_{CLK\text{-}PATH} = \Delta_{CLK\text{-}INT} + \Delta_{CLK\text{-}SYN}$
c)
DATA FLOW
CLK1
CLK2
IN1
CELL 1
Δ_{CELL1}
OUT1
$\Delta_{DATA\text{-}INT}$
$\Delta_{DATA\text{-}SYN}$
IN2
CELL 2
OUT2
$\Delta_{DATA\text{-}PATH} = \Delta_{DATA\text{-}INT} + \Delta_{DATA\text{-}SYN}$

Fig. 1. General idea of counterflow and concurrent flow clocking schemes. a) clock path for counterflow clocking, b) clock path for concurrent flow clocking, c) data path.
Notation is as follows:

$\Delta_{CLK\text{-}INT}$ ($\Delta_{DATA\text{-}INT}$) - delay of interconnections in the clock (data) path;

$\Delta_{CLK\text{-}SYN}$ ($\Delta_{DATA\text{-}SYN}$) - delay of the extra JTL added to the clock (data) path for the purpose of synchronization.

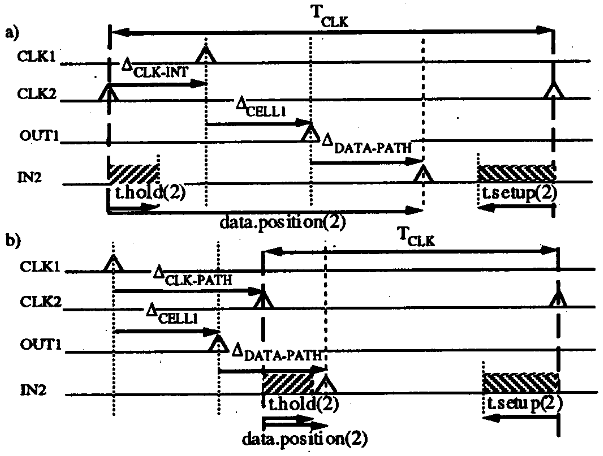

Fig. 2. The exchange of information between neighboring cells in:
a) counterflow clocking scheme, b) concurrent flow clocking scheme.
Notation: Δ_{CELLi} - clock-to-output delay of the cell i;
t.hold(i) - hold time of the cell i; *t.setup(i)* - setup time of the cell i;
data.position(i) - position of the data pulse within clock cycle (cell i).

For counterflow clocking, (1) becomes (see Fig. 2a)

$$t.hold(2) \le \Delta_{CLK\text{-}INT} + \Delta_{CELL1} + \Delta_{DATA\text{-}PATH}, \quad (2)$$

$$T_{CLK} \ge \Delta_{CLK\text{-}INT} + \Delta_{CELL1} + \Delta_{DATA\text{-}PATH} + t.setup(2). \quad (3)$$

Inequality (2) is typically satisfied for $\Delta_{DATA\text{-}PATH} = \Delta_{DATA\text{-}INT}$ (i.e., $\Delta_{DATA\text{-}SYN} = 0$). In this case (henceforth referred to as the counterflow clocking case 1), the minimum clock period in the circuit is

$$T_{CLK}^{MIN} = \Delta_{CLK\text{-}INT} + \Delta_{CELL1} + \Delta_{DATA\text{-}INT} + t.setup(2). \quad (4)$$

When (2) is not satisfied (the counterflow clocking case 2), additional JTL stages must be added in the data path to avoid races ($\Delta_{DATA\text{-}SYN} > 0$).

For concurrent flow clocking, (1) becomes (see Fig. 2b):

$$t.hold(2) \le \Delta_{CELL1} + \Delta_{DATA\text{-}PATH} - \Delta_{CLK\text{-}PATH}, \quad (5)$$

$$T_{CLK} \ge \Delta_{CELL1} + \Delta_{DATA\text{-}PATH} + t.setup(2) - \Delta_{CLK\text{-}PATH}. \quad (6)$$

Extra JTL stages can be added to the clock or data path so that (5) becomes an equality. The minimum clock period of the circuit is therefore

$$T_{CLK}^{MIN} = t.hold(2) + t.setup(2). \quad (7)$$

From (4) and (7) we see that counterflow clocking is usually much slower than concurrent flow clocking, insofar as

$$t.hold(2) < \Delta_{CLK\text{-}INT} + \Delta_{CELL1} + \Delta_{DATA\text{-}INT}. \quad (8)$$

Intuitively, this is because the counterflow data pulse cannot appear at the input of the second cell during the relatively long initial part of the clock period (see Fig. 2a). The clock pulse must propagate to the input of the first cell, which releases data stored in the first cell, thereby permitting the data signal to propagate through the interconnections between the cells. In concurrent flow clocking, this restriction does not apply and the data pulse may appear at the data input of the second cell as early as the hold time after the beginning of the clock cycle (Fig. 2b).

III. Clock Distribution Network Design Accounting For Parameter Variations

A major issue in the design of clock distribution networks is accounting for deviations of circuit parameters due to variations in the fabrication process. These deviations can significantly change the timing parameters of the individual cells and the interconnections within the circuit, and must be taken into account at the system level.

In [11], the timing requirements for some basic RSFQ cells are analyzed in light of the design parameter uncertainties inherent in present day Nb/Al_2O_3/Nb Josephson junction integrated circuit fabrication technologies. It is found that the *relative deviation in the clock-to-output delay,* δ (defined as the 3σ standard deviation in the delay divided by the nominal value of the delay), for most of the basic cells is less than 20%. Also, the absolute deviation in the hold and setup times increases proportionally with δ.

Rewriting (2), (3), (5), (6) to account for these deviations, the worst case conditions for the correct exchange of information between two adjacent cells becomes

- for counterflow clocking

$$t.hold(2)_{MAX} \le \Delta^{MIN}_{CLK-INT}+\Delta^{MIN}_{CELL1}+\Delta^{MIN}_{DATA-PATH}, \quad (9)$$

$$T_{CLK} \ge \Delta^{MAX}_{CLK-INT}+\Delta^{MAX}_{CELL1}+\Delta^{MAX}_{DATA-PATH}+t.setup(2)_{MAX}; \quad (10)$$

- for concurrent flow clocking

$$t.hold(2)_{MAX} \le \Delta^{MIN}_{CELL1}+min\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\}, \quad (11)$$

$$T_{CLK} \ge \Delta^{MAX}_{CELL1}+t.setup(2)_{MAX}+max\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\}. \quad (12)$$

In our concurrent flow clocking scheme

$$\Delta_{CLK-PATH} = \Delta_{CLK-INT}+\Delta_{CLK-SYN}, \quad (13)$$

where $\Delta_{CLK-SYN}$ is the delay of an extra JTL in the clock path (see Fig. 1b), chosen to make (11) an equality for the worst case deviation of all timing parameters.

These conditions lead us to the following general expressions for the minimum clock period for both clocking schemes:

- for counterflow clocking

$$T^{MIN}_{CLK} = t.setup(2)_{MAX}+\Delta^{MAX}_{CELL1}+\Delta^{MAX}_{CLK-INT}+\Delta^{MAX}_{DATA-PATH}, \quad (14)$$

- for concurrent flow clocking

$$T^{MIN}_{CLK} = t.setup(2)_{MAX}+t.hold(2)_{MAX}+(\Delta^{MAX}_{CELL1}-\Delta^{MIN}_{CELL1})+ \\ +(max\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\}-min\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\}). \quad (15)$$

Equations (14) and (15) show the significant advantage of the concurrent flow clocking scheme. The minimum clock period in our scheme no longer directly depends on the clock-to-output delay of the cells but rather depends only on deviations in the delay (i.e., the difference between the maximum and minimum value). Concurrent flow clocking also eliminates the direct dependence of the minimum clock period on the maximum inter-cell connection delay of both data and clock paths, permitting a greater separation between the cells without incurring any degradation in performance.

However, the efficiency of the concurrent flow clocking scheme is greatly affected by the deviation of the timing parameters, as the difference between the maximum and minimum value of any delay changes two times faster as a function of relative delay deviation than the maximum value of the delay.

We further transform (14) and (15) to show their dependence on the nominal values of the timing parameters and their deviations. We assume that both the clock and the data path delays for a given pair of cells are implemented using JTLs located within the same area of the integrated circuit, and thus these deviations are correlated. Thus,

- for $\Delta_{DATA-PATH} - \Delta_{CLK-PATH} < 0$ (henceforth referred to as concurrent flow clocking case 1),

$$max\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\} = \Delta^{MIN}_{DATA-PATH}-\Delta^{MIN}_{CLK-PATH}, \quad (16)$$

- for $\Delta_{DATA-PATH} - \Delta_{CLK-PATH} > 0$ (henceforth referred to as concurrent flow clocking case 2),

$$max\{\Delta_{DATA-PATH}-\Delta_{CLK-PATH}\} = \Delta^{MAX}_{DATA-PATH}-\Delta^{MAX}_{CLK-PATH}. \quad (17)$$

Similar relations hold for the minimum value of the difference between delays. Further transformations lead to the minimum clock period and the extra delay in the clock or data path for both clocking schemes, as summarized in Table I.

One can easily check that under the condition,

$$t.hold(2)_{MAX} \le \Delta^{MIN}_{CLK-INT}+\Delta^{MIN}_{CELL1}+\Delta^{MIN}_{DATA-INT}, \quad (18)$$

concurrent flow clocking always outperforms counterflow clocking. For greater $t.hold(2)_{MAX}$, both schemes offer identical performance.

IV. Design Methodology - How the Equations Are Used

To apply our design strategy for developing RSFQ clock distribution networks, the designer must first establish for each cell in the circuit:

a) the nominal values of the clock-to-output delay, hold time, and setup time;

b) the maximum relative deviation of the clock-to-output delay; and

c) the absolute maximum values of the hold and setup time.

Nominal values of timing parameters can be derived from a circuit level simulator, such as JSPICE or PSCAN. Hold and setup times should be defined to assure correct timing as well as correct functionality: the clock-to-output delay may be affected by too small an interval between the clock and data pulses (see definition of quiescent hold and setup times in [12]). An efficient procedure for establishing deviations of timing parameters is provided in [11]. The maximum of relative delay deviations for all cells is used in further computations as δ.

For all distinct pairs of communicating cells in the circuit under consideration, we must further compute the minimum clock period imposed by the given pair of cells and the extra synchronization delay. All computations can be performed using formulas provided in Table I. The conditions necessary to distinguish between cases 1 and 2 for each clocking scheme are also defined in the table. The largest minimum clock period computed for each pair of cells limits the maximum clock frequency of the circuit. Extra synchronization delay can be implemented using an appropriate number of JTL stages introduced to either the clock path or data path, depending on the sign of the extra delay as specified in Table I.

TABLE I
DESIGN OF CLOCK DISTRIBUTION NETWORKS FOR COUNTERFLOW AND CONCURRENT FLOW CLOCKING SCHEMES

CLOCKING SCHEME	COUNTERFLOW	CONCURRENT FLOW

CONDITION	*if* $t.hold(2)_{MAX} \leq (1-\delta)\left(\Delta^{NOM}_{CELL1} + \Delta^{NOM}_{DATA-INT} + \Delta^{NOM}_{CLK-INT}\right)$ *then* CASE 1
	 else CASE 2
		if $t.hold(2)_{MAX} \leq (1-\delta)\Delta^{NOM}_{CELL1}$ *then* CASE 1
		 else CASE 2

	CASE 1	CASE 1
EXTRA DELAY	No extra delay	$\Delta^{NOM}_{CLK-SYN} - \Delta^{NOM}_{DATA-SYN} = \dfrac{(1-\delta)\Delta^{NOM}_{CELL1} - t.hold(2)_{MAX}}{1+\delta} + \Delta^{NOM}_{DATA-INT} - \Delta^{NOM}_{CLK-INT}$
T^{MIN}_{CLK}	$t.setup(2)_{MAX} + (1+\delta)\left(\Delta^{NOM}_{CELL1} + \Delta^{NOM}_{CLK-INT} + \Delta^{NOM}_{DATA-INT}\right)$	$t.setup(2)_{MAX} + \left(1 - \dfrac{2\delta}{1+\delta}\right)t.hold(2)_{MAX} + \dfrac{4\delta}{1+\delta}\Delta^{NOM}_{CELL1}$
	CASE 2	CASE 2
EXTRA DELAY	$\Delta^{NOM}_{DATA-SYN} = \dfrac{t.hold(2)_{MAX}}{1-\delta} - \Delta^{NOM}_{CELL1} - \Delta^{NOM}_{DATA-INT} - \Delta^{NOM}_{CLK-INT}$	$\Delta^{NOM}_{CLK-SYN} - \Delta^{NOM}_{DATA-SYN} = -\dfrac{t.hold(2)_{MAX}}{1-\delta} + \Delta^{NOM}_{CELL1} + \Delta^{NOM}_{DATA-INT} - \Delta^{NOM}_{CLK-INT}$
T^{MIN}_{CLK}	$t.setup(2)_{MAX} + \left(1 + \dfrac{2\delta}{1-\delta}\right)t.hold(2)_{MAX}$	$t.setup(2)_{MAX} + \left(1 + \dfrac{2\delta}{1-\delta}\right)t.hold(2)_{MAX}$

V. EXAMPLE CIRCUIT

To illustrate the design methodology described in section IV, and to compare the effectiveness of concurrent and counterflow clocking, we apply both schemes to a practical example circuit: the operational unit of the decimation digital filter, first described in [13]. The unit consists of three types of cells: the Parallel Shift Register (PSR), the AND gate (AND), and the Adder-Accumulator (AAC), connected together as shown in Fig.3. The nominal values of the timing parameters for all cells are given in Table II together with the maximum values of the hold and setup times, which correspond to deviations in the Hypres, Inc. technological process, as determined in [11]. There are only four distinct pairs of communicating cells in the filter (see Fig. 3). For each pair of communicating cells, we assume $\Delta_{DATA-INT} = 0$ and $\Delta_{CLK-INT} = 13$ ps (equal to the delay of the splitter plus the delay of the one extra JTL stage). In Table III, we provide the minimum clock period imposed by each pair of cells for each clocking scheme, as well as the corresponding synchronization delays. Both values for both clocking schemes are obtained from the general formulas shown in Table I. In Fig. 4, we present the maximum clock frequency of the filter for counterflow and concurrent flow clocking, compared to the intrinsic speed of the slowest component cell for various delay deviations. The intrinsic minimum clock period of the cell is

$$T^{MIN}_{CLK} = t.setup(2)_{MAX} + t.hold(2)_{MAX}. \qquad (19)$$

We see that concurrent flow clocking has a significant performance advantage over counterflow clocking. For $\delta = 20\%$ (our estimate for fabrication technology currently in use), Table III and Fig. 4 show that concurrent flow clocking offers a speed-up of 89/43=2.1 over counterflow clocking. The minimum clock period of the filter is limited by the

TABLE II
VALUES OF TIMING PARAMETERS FOR THE FILTER COMPONENT CELLS

cell type	clock-to-output delay (ps)	hold time (ps)	setup time (ps)	max. hold time (ps)	max. setup time (ps)
PSR	23	-31	42	-27	46
AND	30	10	-3	12	-1
AAC	24	-13	24	-10	27

TABLE III
MINIMUM CLOCK PERIODS IMPOSED BY EACH PAIR OF COMMUNICATING CELLS (FOR RELATIVE DELAY DEVIATION $\delta = 20\%$)

	COUNTERFLOW		CONCURRENT FLOW	
CELL PAIR	T^{MIN}_{CLK} (ps)	EXTRA DELAY (ps)	T^{MIN}_{CLK} (ps)	EXTRA DELAY (ps)
PSR—PSR	**89**	0	**43**	25
PSR—AND	42	0	22	-8
AND—AAC	79	0	40	15
ADD—AAC	71	0	36	11

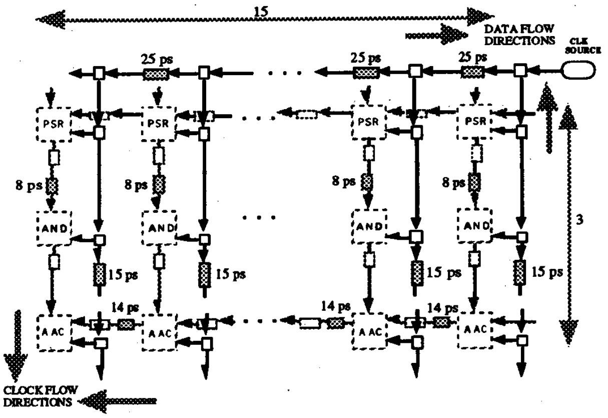

Fig. 3. General structure of the example circuit - operational unit of the decimation digital filter. Clock distribution network corresponds to the concurrent flow clocking scheme. Optimum values of the extra delays in the clock and data paths have been selected according to Table III.

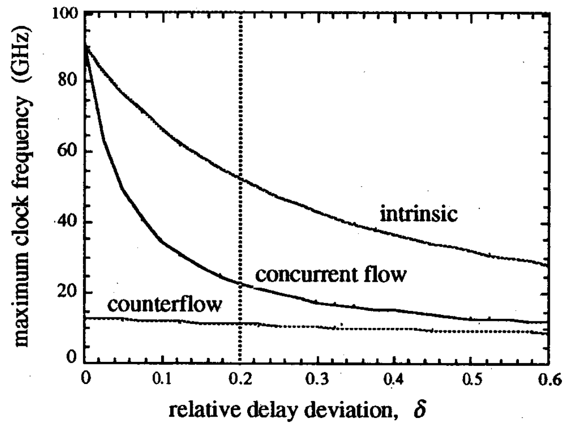

Fig. 4. Maximum frequencies in concurrent flow and counterflow clocking schemes as a function of the relative delay deviation caused by variations in the fabrication process. Results apply to the example circuit, depicted in Fig. 3, with values of timing parameters provided in Table II.

PSR-PSR connection for either clocking scheme (shown as bold entries in Table III). We see that further improvements in the technology for fabricating superconducting digital circuits can yield a dramatic increase in the operational speed of the circuit. For an ideal fabrication technology (δ=0), the concurrent flow clocking takes full advantage of the intrinsic speed of the gates and outperforms counterflow clocking by a factor of seven.

Our theoretical expectations have been confirmed by simulations of the entire operational unit of the decimation filter using the RSFQ logic analyzer — URSULA [13]. Portions of the circuit have also been simulated with JSPICE, giving consistent results. The optimized clock distribution network of the filter is shown in Fig. 3.

This paper has not considered constraints that may be imposed by multiple-input RSFQ cells on the minimum separation time between pulses appearing on two different data inputs of the cell [12]. Note that, one must also be aware of constraints imposed by the exchange of information between the circuit and its environment (e.g., the control unit).

VI. Conclusions

In our paper we present a novel synchronization scheme developed specifically for medium to large scale RSFQ circuits. Concurrent flow clocking is compared with the standard counterflow clocking. Our analysis considers practical issues, such as the deviations of timing parameters due to variations in the fabrication process.

Concurrent flow clocking outperforms conventional counterflow clocking scheme for a large variety of circuit types and a wide range of parameter deviations. The advantage of concurrent flow clocking in RSFQ logic is significantly greater than for semiconductor-based logic. This is due to the following unique features of RSFQ digital electronics:

- In RSFQ circuits, the combinational components of the circuit are combined with storage components (registers) to form simple RSFQ cells. This results in smaller differences between the maximum and minimum delay of the data path between two sequentially-adjacent synchronous cells.
- RSFQ synchronous cells demonstrate excellent correlation between the hold and setup times (to be shown in [11]).
- RSFQ circuits provide well controlled and mutually correlated delays using Josephson transmission lines.

Still, for present day fabrication technologies we show, using a digital filter as an example, only a factor of two performance advantage when using concurrent clocking. This smaller factor is due to the primitive state of today's Josephson junction circuit fabrication technology, primitive in comparison with the many mature commercial semiconductor circuit fabrication technologies. However, as superconductive technologies mature, a dramatic increase in circuit speed with concurrent clock may be expected.

In any case, the performance improvements offered by concurrent flow clocking entail increased logic and layout design complexity. Thus, the choice of clocking scheme for a particular circuit is application-specific. Our paper is intended to provide a sufficient theoretical background for choosing an appropriate clock distribution design methodology for RSFQ circuits.

Acknowledgment

The authors would like to thank Q. P. Herr, S. S. Martinet and Q. Ke for providing optimized designs of RSFQ cells and for many discussions, and the anonymous reviewer for helpful comments.

References

[1] K. K. Likharev, "Rapid Single-Flux-Quantum logic," *The New Superconducting Electronics*, pp. 423-452, Kluwer Academic Publishers, 1993.

[2] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz-clock frequency digital systems," *IEEE Trans. Appl. Superconduct.*, vol. 1, pp. 3-28, March 1991.

[3] O. A. Mukhanov, S. V. Rylov, V. K. Semenov, and S. V. Vyshenskii, "RSFQ logic arithmetic," *IEEE Trans. Magn.* vol. 25, pp. 857-860, March 1989.

[4] J. Fleischman and T. Van Duzer, "Computer architecture issues in superconductive microprocessors," *IEEE Trans. Appl. Superconduct.* vol. 3, pp. 2716-2719, March 1993.

[5] M. Afghahi and C. Svensson, "Performance of synchronous and asynchronous schemes for VLSI systems," *IEEE Trans. Computers*, vol. 41, pp. 858-872, July 1992.

[6] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*, IEEE Press, 1995.

[7] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Computers*, vol. 39, pp. 945-951, July 1990.

[8] J. L. Neves and E. Friedman, "Circuit synthesis of clock distribution networks based on non-zero clock skew," *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 4.175-4.178, June 1994.

[9] O. A. Mukhanov, "Rapid Single Flux Quantum (RSFQ) shift register family," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 2578-2581, March 1993.

[10] C. A. Hamilton and K. C. Gilbert, "Margins and yield in Single Flux Quantum logic," *IEEE Trans. Appl. Superconduct.*, vol. 1, pp. 33-38, Dec. 1993.

[11] K. Gaj, Q. Herr, and M. Feldman, "Parameter variations and synchronization of RSFQ cells," unpublished.

[12] K. Gaj, Q. Herr, and M. Feldman, "Analysis of timing requirements for basic RSFQ cells," unpublished.

[13] A. Krasniewski, "Logic simulation of RSFQ circuits," *IEEE Trans. Appl. Superconduct.*, vol. 3, pp. 33-38, March 1993.