CHOICE OF THE OPTIMUM TIMING SCHEME FOR RSFQ DIGITAL CIRCUITS[›]

Kris Gaj, Eby G. Friedman, and Marc J. Feldman
Department of Electrical Engineering, University of Rochester, Rochester, New York 14627

ABSTRACT

Rapid Single Flux Quantum (RSFQ) logic is a superconducting digital circuit technology that has emerged as a possible alternative to advanced semiconductor technologies for large scale ultra-high speed, very low power digital applications. Timing of RSFQ circuits at frequencies of tens to hundreds of gigahertz is a challenging and still unresolved problem. In many circuit topologies of current medium to large scale RSFQ circuits synchronous clocking outperforms asynchronous schemes in speed, device/area overhead, and simplicity of the design procedure. Circuit yield and performance of large synchronous circuits is limited by process parameter variations. As a result, alternative synchronization techniques, including but not limited to asynchronous timing, should be considered for certain circuit topologies. We discuss our methodology for the optimal choice of the timing scheme for medium to large RSFQ circuits.

INTRODUCTION

The recent achievements of superconductive circuits using *Rapid Single Flux Quantum (RSFQ)* logic make this technology a possible candidate to first cross the boundary of 100 GHz clock frequency in a large scale digital circuit [1]. Correct timing is essential to fully exploit the high speed capability of individual RSFQ gates, and to translate this advantage into a corresponding speed-up in the performance of medium to large scale RSFQ circuits. Research in this area has only just started and has been only applied to moderate 100-gate circuits to date. Yet even for this medium scale complexity, the design of effective timing schemes in the multi-gigahertz frequency range is a challenging problem [2].

The timing of medium to large scale RSFQ circuits is governed by the same rules and constraints as the timing of large scale VLSI semiconductor-based circuits. Therefore, one approach to superconducting circuit design is to rely on the well-established principles and methodologies of VLSI timing, and to apply the design rules and techniques drawn from the semiconductor literature. There are however significant differences which arise from a) the lack of purely combinational logic in RSFQ circuits; b) the low fanout of RSFQ gates; c) a different suite of elementary gates in the two technologies. As a result, the RSFQ clocking circuitry is usually developed specifically for RSFQ logic [3, 4].

A central dilemma is the choice between synchronous and asynchronous clocking. RSFQ logic is well suited for both types of clocking. Asynchronous timing requires local signaling between adjacent cells. This signaling is naturally based on the concept of *events* such as *request* and *acknowledge*. In semiconductor logic events are coded using voltage state *transitions*. Semiconductor logic elements that process voltage transitions are complex and slow compared to logic gates that process *voltage levels*. As a result, asynchronous timing has *not* been widely accepted in semiconductor circuit design. In RSFQ logic, events are coded using SFQ pulses. Asynchronous logic elements that process SFQ pulses are simple and fast, and therefore RSFQ asynchronous circuits can approach the speed of synchronous circuits. Therefore, asynchronous event driven schemes such as *dual-rail logic* [5] or *micropipelines* [6] appear to be easier and more natural to implement in RSFQ circuits than in semiconductor-based logic. The advantages of asynchronous clocking include the larger tolerance to fabrication process variations and circuit modularity; its disadvantages: device/area overhead, as well as complicated design and testing procedures.

Because of the high speed of RSFQ circuits, many have argued that some asynchronous scheme *must* be used, because any synchronous clocking (using single global clock signal) will certainly be inadequate for any very high speed circuit. Nevertheless, synchronous clocking has been successfully used in almost all medium to large scale RSFQ circuits developed to date.

[›] This work was supported in part by the Rochester University Research Initiative sponsored by the US Army Research Office.

Its primary advantages include high performance, design simplicity, small device and area overhead, and good testability.

Synchronous clocking works well for RSFQ circuits even at very high speed because RSFQ circuit designs almost always ignore the well-established clocking technique used in most semiconductor circuits. This is equipotential zero-skew clocking, in which every gate is clocked at the same instant, before the clock source generates its next output. In RSFQ logic the timing signals are SFQ pulses which pass through the clock network similarly to the data flow. It is then natural to choose pipelined clocking, either counterflow or concurrent, where many clock SFQ pulses travel through the clock path in parallel to the data path. These non-zero-skew clocking schemes are chosen either for superior performance or for extended tolerance to fabrication process variations [2].

CHOICE OF THE CLOCKING SCHEME

The choice of the clocking scheme for a particular RSFQ circuit is influenced by:
a) the topology of the circuit (e.g., one-dimensional vs. two-dimensional array, regular vs. irregular structure);
b) the performance requirements (throughput, latency) of the circuit;
c) global and local parameter variations in the circuit;
d) complexity of the design procedure (computationally intensive Monte Carlo analysis vs. analytical estimations);
e) the device, area, and power consumption overhead;
f) the complexity of the physical layout.

For circuits which are roughly linear (pipeline width << length), e.g., for one-dimensional Nx1 arrays, and NxM arrays with large N and small M, the natural choices are the *straight-line* synchronous clocking schemes in which the clock path is distributed in parallel to the main data path of the array. Two types of straight-line clocking can be distinguished. In *counterflow clocking*, the clock flows in the direction *opposite* to the data. In *concurrent clocking* the clock and the data flow in the *same* direction. *Counterflow clocking* offers the advantages of high robustness to timing parameter variations, small area, and a simple design procedure, but at the cost of reduced circuit throughput. When the highest clock frequency is of primary concern, *concurrent clocking* should be considered. An aggressive application of this scheme will reduce the expected yield of the circuit unless there is a good quantitative knowledge of the fabrication process variations. The design procedure leading to the optimum solution may require intensive Monte Carlo simulations, although suboptimal solutions can be obtained using simpler analytical methods [4]. Concurrent clocking tends to require a larger number of Josephson transmission line (JTL) stages in the clock paths compared to counterflow clocking, and thus a greater overhead in circuit area and in layout complexity is expected.

Asynchronous schemes such as *dual-rail clocking* [5] and *micropipelines* [6] can also be successfully applied to linear and asymmetric arrays, but these schemes do not offer any advantages over synchronous schemes in either performance, robustness, or design complexity. Either scheme can be adjusted (by the appropriate choice of interconnect delays) to provide either the performance equivalent of concurrent clocking or the robustness of counterflow clocking. Both schemes, however, require a significant overhead. The design for optimum performance is equally complex as for concurrent clocking and requires good knowledge of the timing parameter variations. Generally, for both synchronous and asynchronous schemes, the maximum clock frequency in linear (or roughly linear) arrays is independent of the array length.

For a two-dimensional symmetric square NxN arrays the situation is more complicated. The effects of the local parameter variations in the paths of the clock distribution network cause additional clock skew, and as a result reduce circuit performance and yield. In all of the synchronous schemes, the performance of the circuit deteriorates by a factor proportional to at least $\sqrt{N}$. Depending on the magnitude of the on-chip variations and the topology of the clock distribution network, these effects may become critical for different sizes of N. In particular, it is possible that these effects may be sufficiently small for practical sizes of RSFQ arrays, especially for counterflow clocking. For all synchronous schemes, the worst case maximum clock frequency deteriorates with increasing N. Additionally, for all single-phase clocking schemes, there exists a value of N above which the low-speed yield of the circuit begins to decrease. This value of N is smallest for concurrent clocking and largest for counterflow clocking.

Asynchronous schemes scale better with increasing N, as the timing signals are generated locally and do not need to travel through the long branches of the clock distribution network. Again, the primary disadvantage of these schemes is the large circuit overhead. These schemes are also more difficult to analyze and test than synchronous schemes. As a result, the use of asynchronous timing methodologies may be limited to circuits of large N. Finally, hybrid synchronization schemes which use asynchronous strategies in tandem with simpler synchronous schemes are likely to be advantageous for large RSFQ circuits.

A novel synchronous timing scheme which addresses the square array problem is resynchronized clocking, used for a parallel multiplier in [3]. The additional clock skew is eliminated using special RSFQ gates called coincidence junctions. Another potentially valuable scheme is two-phase synchronous clocking proposed in [2]. This scheme is expected to offer better performance than concurrent clocking, better tolerance to fabrication process variations than counterflow clocking, and an extremely simple design procedure. Also, in two-phase clocking, the choice of the optimum interconnects in the circuit does not require any knowledge of the timing parameter variations. Increasing the array size N does not influence the choice of interconnect delays, and deteriorates only the *worst case* circuit performance. Neither the *expected* performance of the circuit nor the functional circuit yield at low speed is affected by an increase of the array size. Both schemes require further research to be optimized for speed and robustness, and to minimize the area/device overhead they introduce.

timing scheme	sync/ async	speed	robustness	design procedure simplicity	area overhead	suitable for linear Nx1 arrays	suita sym NxN
counterflow	S	–	+	++	+	+	+/–
concurrent	S	+	–	––	–	+	–
zero-skew	S	–/+	–	+	–/+	+	–
resynchronized	S	–	++	+	––	+	+
two-phase	S	++	++	+	––	+	+
dual-rail	A	+/–	+	–	––	+	+
micropipelines	A	–	+	–	––	+	+/–

REFERENCES

[1] K. Likharev, "Recent Progress and Prospects of Superconductor Digital Technology," FED Report, January 1997.

[2] K. Gaj, E.G. Friedman, and M.J. Feldman, "Timing of Multi-Gigahertz Rapid Single Flux Quantum Digital Circuits," Journal of VLSI Signal Processing, vol. 9, 1997 (in print).

[3] O.A. Mukhanov, S.V. Rylov, V.K. Semenov, and S.V. Vyshenskii, "RSFQ Logic Arithmetic," *IEEE Trans., Magnetics*, vol. 25, 1989, pp. 857-860.

[4] K. Gaj, E.G. Friedman, M.J. Feldman, and A. Krasniewski, "A Clock Distribution Scheme for Large RSFQ Circuits," *IEEE Trans. Appl. Supercond.*, vol. 5, 1995, pp. 3320-3324.

[5] M. Maezawa, I. Kurosawa, Y. Kameda, and T. Nanya, "Pulse-Driven Dual-Rail Logic Gate Family Based on Rapid Single-Flux-Quantum (RSFQ) Devices for Asynchronous Circuits," *Proc. 2nd Int. Symposium Advanced Research in Asynchronous Circuits and Systems*, March 1996, pp. 134-142.

[6] I.E. Sutherland, "Micropipelines," *Comm. ACM*, vol. 32, 1989, pp. 720-738.