

RSFQ LOGIC ARITHMETIC

O.A. Mukhanov⁺, S.V. Ryllov^{*}, V.K. Semenov^{*}, and S.V. Vyshenskii⁺.

(+) Institute of Nuclear Physics, and (*) Department of Physics, Moscow State University, Moscow 119899, U.S.S.R.

Abstract

Several ways of local timing of the Josephson-junction RSFQ (Rapid Single Flux Quantum) logic elements are proposed, and their peculiarities are discussed. Several examples of serial and parallel pipelined arithmetic blocks using various types of timing are suggested and their possible performance is discussed. Serial devices enable one to perform n-bit functions relatively slowly but using integrated circuits of a moderate integration scale, while parallel pipelined devices are more hardware-wasteful but promise extremely high productivity.

Introduction

Progress in development of the Josephson junction digital devices was traditionally associated with latching logic elements. These elements, however, do have several drawbacks including relatively long (~1 ns) reset time which prevents clock frequencies above the 1 GHz threshold already crossed by fast semiconductor integrated circuits.

Recently a new "RSFQ" (Resistive or Rapid Single Flux Quantum) family of nonlatching logic elements employing overdamped Josephson junctions has been proposed¹ and analysed². The analysis² and the first experiments^{3,4} have shown that the basic members of the family can operate at extremely high clock frequencies: at least 30 GHz for 10-μm Nb technology^{3,4} and up to 500 GHz for 1-μm Nb technology².

Nevertheless, specific problems arising in the digital devices operating at frequencies so high, notably the problems of connections and timing of the logic elements, have not been analysed so far. The purpose of the present work was to show that structure of the RSFQ logic allows natural solutions of those two major problems.

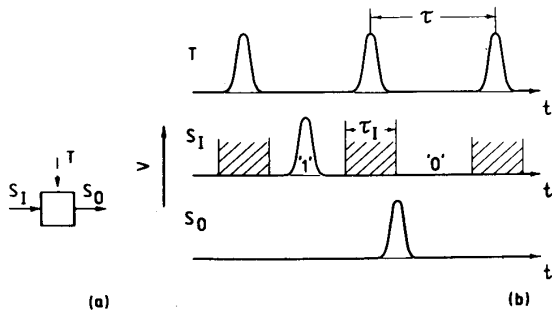


Figure 1. Representation of information in the RSFQ logic circuits: (a) general structure of an RSFQ logic element, and (b) time diagram illustrating mutual time arrangement of the RSFQ clock pulses T, input pulses S₁, and output pulses S₀. τ_I is the part of the clock period forbidden for arrival of the input pulses.

ABC of the RSFQ Circuitry

In contrast with latching logic elements, those of the RSFQ logic family exchange information bits in form of single flux quanta, i.e. of short (picosecond) voltage pulses V(t) with the area ∫ V(t)dt = φ₀. Binary unity/zero is presented by presence/absence of a SFQ pulse in a signal line S during the time period between two consequent SFQ pulses in a clock (timing) line T - see Fig.1 (in most figures of this paper the clock lines are shown dashed).

Each logic element of the RSFQ family contains one or several interferometers with two stable states which differ by single flux quantum. Input signal pulse(s) S_I can switch these interferometers, so that their final state corresponds to a logic function being performed by the element. Clock pulse T fed into special input of the element (Fig.1) resets the interferometers into their initial state and simultaneously initiates delivery of the output SFQ pulses at the element output terminal(s) S_O.

Figures 2-4 show the basic logic elements which we will need for our discussion. Two of them, the DRO register cell R and full adder FA (Fig.2) have been discussed in detail earlier², so that we will describe operation of only two new elements.

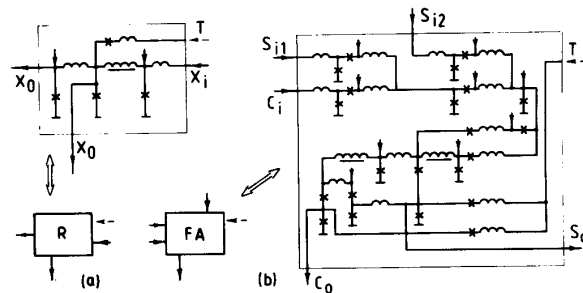


Figure 2. Basic RSFQ logic elements: (a) DRO register cell R; (b) 1-bit full adder FA.

NDRO/DRO Register Cell N (Fig.3). This element presents a slight complication of that shown in Fig.2a. Its terminals Y_I and Y_O serve for writing down and reading out the cell contents, similarly to X_I and X_O terminals in Fig.2a (this operation is initiated by the pulse T). Except that the cell has an additional output terminal P for non-destructive reading out which can be initiated by the pulse arriving at the terminal X. Within the RSFQ coding system, output P corresponds to the logic function P = X·Y_I.

Coincidence Junction C (Fig.4) is used exclusively in clock line structures, because its dynamics suggests that the pulses T₁ and T₂ arriving at its inputs are always bound in pairs, although with arbitrary time shift. Upon arrival of the last bit of the pair the element produces the output pulse T₃ and resets to its initial state.

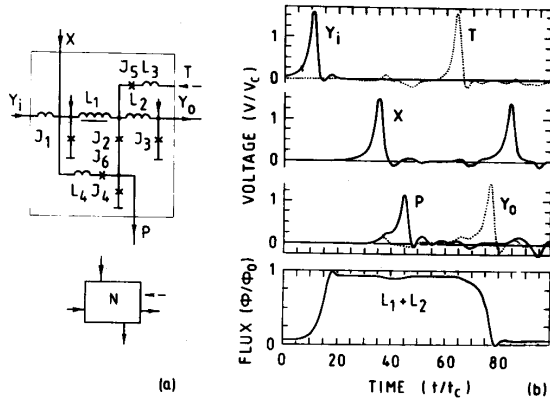


Figure 3. NDRO/DRO register cell N: (a) equivalent circuit and notation; (b) results of numerical simulation of the cell with parameters: $V_c = 0.3$ mV, $t_c = 1.1$ ps, $\beta_c = 1$, $I_{c3} = 2I_{c2} = 2I_{c4} = 1.33I_{c1} = 1.33I_{c5} = 1.33I_{c6} = 0.25$ mA, $L_1 = 1.8L_2 = 3.5L_4 = 3.5L_4 = 12$ pH.

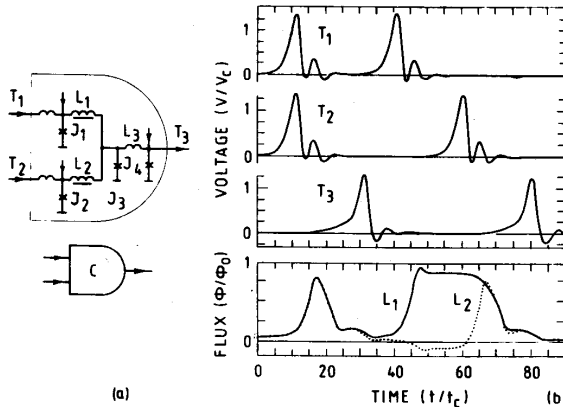


Figure 4. Coincidence Junction C: (a) equivalent circuit and notation; (b) results of numerical simulation for parameters: $V_c = 0.3$ mV, $t_c = 1.1$ ps, $\beta_c = 1$, $2I_{c1} = 2I_{c2} = 2I_{c3} = I_{c4} = 0.25$ mA, $L_1 = L_2 = L_3 = 16$ pH.

Timing

Global timing used in particular in Josephson junction latching logics becomes impossible in ten-hundred-GHz clock-frequency range. This is why the RSFQ logic circuits should operate in local timing mode where T pulses arrive at a logic element from one of its direct neighbours, either the information consumer (Fig. 5a), or from its producer (Fig. 5b), or both (Fig. 5c).

In order to feel the difference of the two simplest ways of timing, consider one-dimensional arrays of elements shown in Fig.5a,b. In the former case, a single clock pulse T fed into the array edge will cause a shift of all signal bits by one elementary cell. On

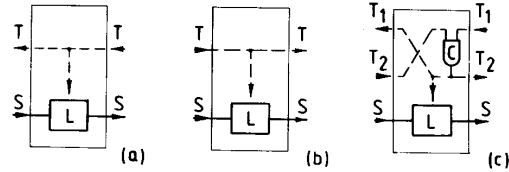


Figure 5. Three possible ways of timing the RSFQ logic elements: (a) and (b) timing suitable for simple circuits; (c) self-timing appropriate for more complex circuits.

the contrary, in the latter case the pulse will drive a single bit along all the array (if logic delays of the signal lines are less than those of clock lines). The both ways, as well as their combinations, can be successfully used in relatively simple digital circuits (for example, see below). They do not, however, defend the circuits against random delays of the clock pulses, which are unavoidable in more complex devices. The problem can be completely solved with the timing circuit shown in Fig.5c. One can readily get convinced that the circuit provides just the same result as one of those shown in either Fig.5a or Fig.5b, depending on the initial setting of the C circuits. However, in contrast with its simpler counterparts, the timing circuit shown in Fig.5c automatically delays the clock pulses if they arrive too early. If one supplies a total digital device by a generator of initial clock pulses (say, T_1) also controlled by the complementary pulse train (T_2) we obtain a completely self-timed device which automatically operates at the highest speed permitted by single elements (i.e. tens or even hundreds GHz for the RSFQ logic).

Arithmetic Blocks

As an illustration of various methods of timing let us consider possible structure of arithmetic blocks performing either serial or parallel bit-level processing of input data.

Serial Multiplication. Figure 6 shows a serial multiplier (SM) of two n-bit numbers (X,Y) composed of the logic elements discussed above. It is controlled by two independent clock trains T_x and T_y . Firstly, the T_y train loads bits of Y to the register of NDRO/DRO cells R. Then starts the train T_x which induces consequent motion of bits of X through the register of DRO cells R and simultaneously the backward motion of the bits through the register of full adders FA. One can

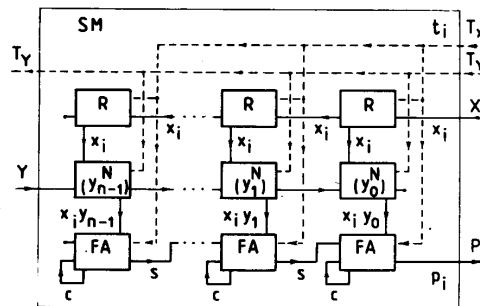


Figure 6. $n \times n$ -bit serial multiplier with save carry and shift sum.

easily check up that each clock period the device produces a correct consequent bit of the 2n-bit product XY at its output P, so that the whole operation cycle takes 2n clock periods (loading of new Y is supposed to be fulfilled during last n periods of the previous cycle).

Note that we have used the timing scheme shown in Fig.5a in two lower rows (registers) of the multiplier while that shown in Fig.5b is used in its upper row. For large n, application of the combined timing scheme (Fig.5c) can become necessary.

Serial Division. Figure 7a shows an arithmetic block D_j for serial calculation of the n-bit reciprocal $Z = 1/M$ ($1/2 < M < 1$) using $k = \log_2 n$ iterations

$$Z_j = Z_{j-1} (2 - M Z_{j-1}) \quad (1)$$

starting with $Z_0 = 2 - M$ (each iteration doubles the number of correct bits). The block consists of two serial multipliers SM (Fig.6) and a simple device "2-" calculating difference between 2 and an input number (the latter device consists of just two single-bit logic elements, inverter and full adder); the pipelined structure of the block provides the cycle time of one

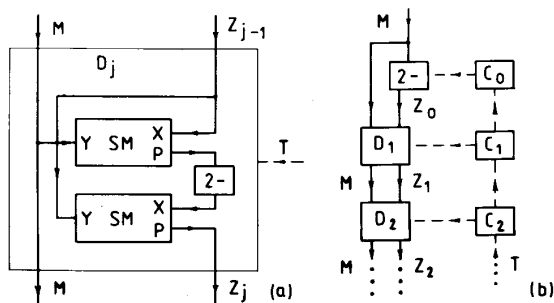


Figure 7. n-bit 1/M divider: (a) an arithmetic block D_j for calculation of the consequent iteration Z_j ; (b) pipeline divider consisting of the blocks D_j and clock controllers C_j .

iteration somewhat less than that required for two multiplications.

The complete divider can be arranged in two different ways: the simplest one is to link the Z_{j-1} and Z_j terminals of a single D block by the shift register and to run it during $\log_2 n$ cycles, while the other one is to use a pipelined structure (Fig.7b) composed of $\log_2 n$ blocks D_j with increasing bit lengths ($n_j \approx 2^j$). The total division time for these two cases is $\sim 2n \log_2 n$ and $4n$ clock periods, respectively.

Parallel Multiplication. Productivity of calculations can be increased drastically using parallel pipelined single-bit units. Figure 8 shows an example of such a device, a multiplier of n-bit numbers X and Y. It presents a two-dimensional array (Fig.8b) of single bit units, either those shown in Fig.8a or more simple logic cells R (Fig.2a) and half-adders A. One can see that the basic unit shown in Fig.8a is a copy of a column unit of the serial multiplier (Fig.6); the only difference is a way of connection of the units (Fig.8b) and their timing (Fig.8c). The clock pulse distribution network (Fig.8c) using coincidence junctions C (Fig.4) ensures upward motion of the horizontal wave fronts of the T pulses preventing the front skew (for more complex devices, combination of the timing circuits shown in Fig.5c and Fig.8c can be necessary). The T front induces downward motion of horizontal wave fronts of the signal bits through the multiplier array. When fed in parallel by all 2n bits of new operands X, Y each clock period, the multiplier produces 2n bits of the product $P = XY$ each period, although processing of a given pair of operands takes $\sim 2n$ clock periods.

Discussion

Now we can discuss possible structure of digital integrated circuits using the RSFQ arithmetic blocks. Here one should take into account that the picosecond RSFQ pulses can hardly be passed between the integrated circuits because of relatively large frequency dispersion of the interchip connectors. It means that the RSFQ bits produced by the RSFQ circuit should be at first converted to the usual dc voltage ("potential") form, then passed to another chip, and lastly converted again to the RSFQ form. The converters of the both types has been successfully tested recently;⁶ their delay time is apparently well below 0.1 ns, but the whole chip-to-chip communication can hardly be fulfilled in less than 0.5 ns.

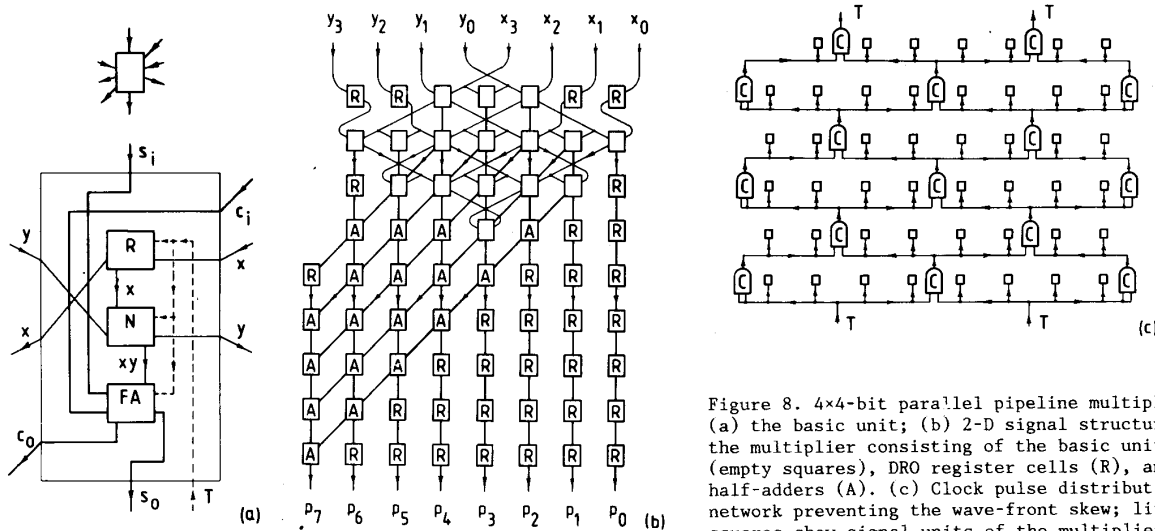


Figure 8. 4x4-bit parallel pipeline multiplier: (a) the basic unit; (b) 2-D signal structure of the multiplier consisting of the basic units (empty squares), DRO register cells (R), and half-adders (A). (c) Clock pulse distribution network preventing the wave-front skew; little squares show signal units of the multiplier.

With this factor taken into account, one arrives at the possible structure shown in Fig.9. The clock controller initiated by external acknowledgement signal generates a train of the SFQ clock pulses (T_1 in Fig.5c); the generation speed is controlled by the backward stream of the pulses T_2 . This train controls the parallel/serial and serial/parallel converters and a serial RSFQ device (of course, parallel processing can be used inside device as well).

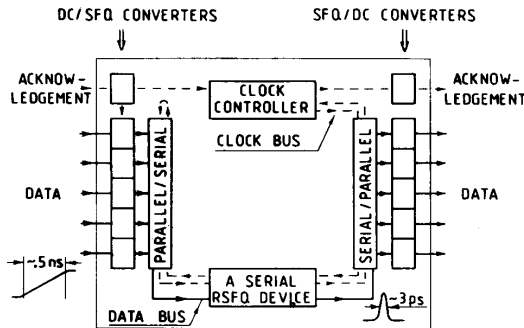


Figure 9. Possible structure of the RSFQ-logic digital integrated circuit.

Due to extremely high clock frequencies of the RSFQ logic elements, all components of the circuit shown in Fig.9 can be quite matched in their speed. For example, the device shown in Fig.6, fabricated using $1\text{-}\mu\text{m}$ technology, could multiply two 64-bit numbers in ~ 0.7 ns, the period quite comparable with conversion and inter-chip communication delays. Note that integration scale

Table 1. Estimates of basic parameters of 32×32 -bit fixed-point multipliers based on various technologies.

Device type; technology	Integration scale, Josephson/p-n junctions	Productivity, operations per second	Time delay, ns
Serial; RSFQ ($2.5\ \mu\text{m}$)	1,500	0.5×10^9	2
Parallel pipelined; RSFQ ($2.5\ \mu\text{m}$)	40,000	30×10^9	2
Parallel; latching ($2.5\ \mu\text{m}$)	70,000	0.5×10^9	2
Parallel; GaAs ($0.5\ \mu\text{m}$)	100,000	0.15×10^9	6
Parallel pipelined; Si-MOS ($1.0\ \mu\text{m}$)	200,000	0.2×10^9	150

of such circuit ($\sim 2 \times 10^3$ Josephson junctions) would be much less than that of a semiconductor (GaAs) chip with the similar performance ($\sim 10^5$ transistors).

On the other hand, using internally-parallel RSFQ devices which require larger integration scales one could attain extremely high productivities of single-chip ICs, which can hardly be approached by semiconductor microelectronics in a foreseeable future (Table 1). Just one example of such device is a ripple-counter-type A/D converter with digital filtering of the output signal⁸.

Conclusion

The main result of our work is a proof of possible local and self-timing of the RSFQ logic elements enabling one to compose digital blocks and complex devices operating at extremely high clock frequencies limited only by logic delays of the RSFQ elements (~ 100 GHz for the present-day Nb technologies). We believe that this fact justifies further intensive development of the RSFQ logic circuits, and makes prospects of latching logic rather doubtful.

Acknowledgement

Usefull discussion with K.K. Likharev, A.T. Rakhimov, and N.N. Roi is gratefully acknowledged.

References

- [1] K.K. Likharev, O.A. Mukhanov, and V.K. Semenov, "Resistive Single Flux Quantum Logic for the Josephson junction digital technology", In: "SQUID'85", Berlin: Walter de Gruyter, pp.1103-1108, 1985.
- [2] O.A. Mukhanov, V.K. Semenov, and K.K. Likharev, "Ultimate Performance of the RSFQ Logic Circuits", *IEEE Trans. Magn.*, vol. MAG-23, pp. 759-762, March 1987.
- [3] V.P. Koshelets, K.K. Likharev, V.V. Migulin, O.A. Mukhanov, G.A. Ovsyannikov, V.K. Semenov, I.L. Serpuchenko, and A.N. Vystavkin, "Experimental Realization of a Resistive Single Flux Quantum Logic Circuit", *IEEE Trans. Magn.*, vol. MAG-23, pp. 755-758, March 1987.
- [4] V.K. Kaplunenko, V.P. Koshelets, K.K. Likharev, V.V. Migulin, O.A. Mukhanov, G.A. Ovsyannikov, V.K. Semenov, I.L. Serpuchenko, and A.N. Vystavkin, "Experimental Study of the RSFQ Logic Circuits", In: "Extended Abstracts of ISEC'87", Tokyo, pp. 127-130, 1987.
- [5] S. Kotani, N. Fujimaki, S. Morohashi, S. Ohara, and S. Hasuo, "Feasibility of an Ultra-High-Speed Josephson Multiplier", *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 98-103, January 1987.
- [6] V.K. Kaplunenko, M.I. Khabipov, V.P. Koshelets, K.K. Likharev, O.A. Mukhanov, V.K. Semenov, I.L. Serpuchenko, and A.N. Vystavkin, "Experimental Study of the RSFQ Logic Elements", presented at this conference.
- [7] K. Gono, I. Honbori, M. Wada, K. Togashi, and Y. Kato, "A GaAs 8x8-bit Multiplier Using JFET DCFL", *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 523-529, August 1986.
- [8] S.V. Rylov, V.K. Semenov, and K.K. Likharev, "Josephson junction A/D converter using differential coding", *IEEE Trans. Magn.*, vol. MAG-23, pp.735-738, March 1987.