Addition Related Arithmetic Operations via Controlled Transport of Charge

Sorin Cotofana, *Senior Member*, *IEEE*, Casper Lageweg, *Student Member*, *IEEE*, and Stamatis Vassiliadis, *Fellow*, *IEEE*

Abstract—This paper investigates the Single Electron Tunneling (SET) technology-based computation of basic addition related arithmetic functions, e.g., addition and multiplication, via a novel computation paradigm, which we refer to as electron counting arithmetic, that is based on controlling the transport of discrete quantities of electrons within the SET circuit. First, assuming that the number of controllable electrons within the system is unrestricted, we prove that the addition of two n-bit operands can be computed with a depth-2 network composed out of $3n+1$ circuit elements and that the multiplication of two n-bit operands can be computed with a depth-3 network composed out of $4n-1$ circuit elements. Second, assuming that the number of controllable electrons cannot be higher than a given constant r determined by practical limitations, we prove that the addition of two n-bit operands can be computed with a $\text{depth-}\left(\frac{n}{r}+3\right)$ network composed out of $3n+1+\frac{n}{r}$ circuit elements. Under the same restriction, we suggest methods to reduce the addition network depth in the order of $\log\frac{n}{r}$ and to perform n-bit multiplication in an $O(\log\frac{n}{r})$ delay. Finally, we propose SET-based implementations for a set of basic electron counting building blocks and implement a number of circuits operating under the electron counting paradigm as follows: 4-bit Digital to Analog Converter, 5-bit Analog to Digital Converter, 4-bit adder, and 3-bit multiplier. All proposed implementations are verified by means of simulation.

Index Terms—SET, single electron technology, electron counting, addition, multiplication.

1 Introduction

Feature size reduction in microelectronic circuits has been an important contributing factor to the dramatic increase in the processing power of arithmetic circuits. However, it is generally accepted that, sooner or later, MOS-based circuits cannot be reduced further in (feature) size due to fundamental physical restrictions [1]. Therefore, several emerging technologies are currently being investigated [2]. Single Electron Tunneling (SET) [3] is one such technology candidate and offers greater scaling potential than MOS, as well as ultra-low power consumption. Additionally, recent advances in silicon-based fabrication technology (see, for example, [4]) show potential for room temperature operation. However, similar to other future technology candidates, SET devices display a switching behavior that differs from traditional MOS devices and this provides new possibilities and challenges for implementing digital circuits.

SET technology introduces the quantum tunnel junction as a new circuit element for (logic) circuits. The tunnel junction can be thought of as a "leaky" capacitor such that the "leaking" can be controlled by the voltage across the tunnel junction. Although this behavior at first glance appears similar to that of a diode, the difference stands in the scale at which switching occurs. Charge transport though a tunnel junction can only occur in quantities of a single electron at a time. Additionally, given the feature sizes anticipated for such circuits, the transport of a single electron can have a significant effect on the voltage across a tunnel junction such that transporting a few electrons through a tunnel junction will inhibit further charge transport, making it possible to control the transport of charge in discrete and accurate quantities.

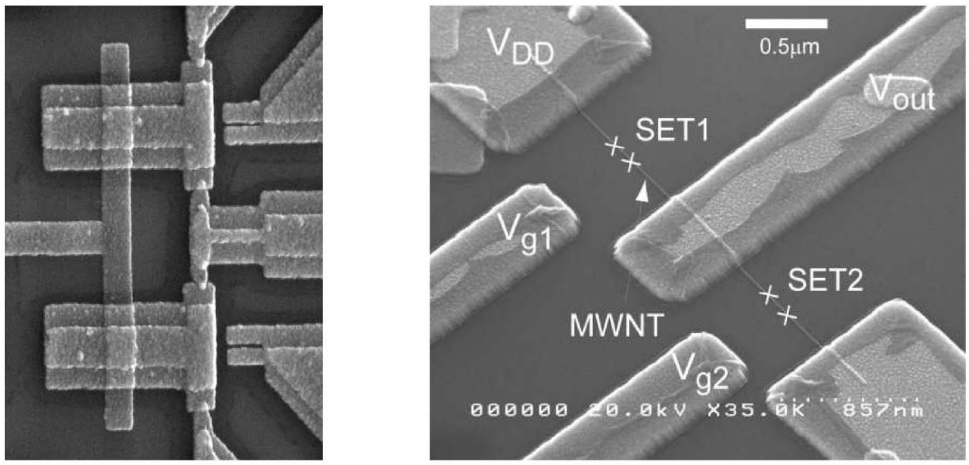
The ability to control the transport of individual electrons in SET technology introduces a broad range of new possibilities and challenges for implementing computer arithmetic circuits. In this paper, we introduce a new computation paradigm, which we refer to as electron counting arithmetic, that is based on controlling the transport of discrete quantities of electrons within the SET circuit. First, we propose a basic set of electron counting building blocks, i.e., move charge block ($MVke$) and periodic symmetric function block (PSF). We subsequently propose electron counting-based schemes for computing addition and multiplication and prove that the following holds true:

- When the number of the electrons that can be accurately controlled within the system is unrestricted, the addition/subtraction of two n-bit operands can be computed with a depth-2 network composed out of $3n+1$ circuit elements.[1] The multiplication of two n-bit operands can be computed with a depth-3 network with $4n-1$ circuit elements.
- When the number of the electrons that can be accurately controlled by an $MVke$ block is limited to 2^r-1, the addition/subtraction of two n-bit operands can be computed with a $\text{depth-}\left(\frac{n}{r}+3\right)$ network composed out of $3n+1+\frac{n}{r}$ circuit elements.

1. By circuit element, we mean in this context any of the building block presented in Section 3.

• *The authors are with the Computer Engineering Laboratory, Delft University of Technology, Delft, The Netherlands.*
E-mail: {Sorin, Casper, Stamatis}@ce.et.tudelft.nl.

Manuscript received 1 Dec. 2003; revised 17 July 2004; accepted 17 Sept. 2004; published online 18 Jan. 2005.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TCSI-0252-1203.



(a) (b)

Fig. 1. SET inverter implementations. (a) Conventional process. (b) Carbon nanotube.

- Under the same restriction, we suggest methods to reduce the addition network depth in the order of $\log\frac{n}{r}$ and to perform n-bit multiplication in an $O(\log\frac{n}{r})$ delay.

Additionally, we propose SET-based implementations for the $MVke$ and PSF blocks. We validate the proposed structures by means of simulation, evaluate their expected performance, and subsequently utilize them in constructing a number of circuits operating under the electron counting paradigm as follows: 4-bit Digital to Analog Converter, 5-bit Analog to Digital Converter, 4-bit adder, and 3-bit multiplier.

The remainder of this paper is organized as follows: Section 2 briefly presents the SET background theory. In Section 3, we discuss a set of charge transport building blocks and proposes schemes for charge transport-based addition and multiplication. Section 4 discusses the implications of practical limitations on the structures described in Section 3 and proposes alternative solutions that can operate under such limitations. Section 5 introduces possible SET-based implementations of the charge transport building blocks and Section 6 presents a number of examples. Finally, Section 7 concludes the paper.

2 Background

Single Electron Tunneling technology introduces the quantum tunnel junction as a new circuit element. A tunnel junction consists of two conductors separated by an extremely thin insulating layer. The insulating layer acts as an energy barrier which inhibits charge transport under normal (classical) physics laws. However, according to quantum physics theory, charge transport of individual electrons through this insulating layer can occur if this results in a reduction of the total energy present in the circuit. The transport of charge through a tunnel junction is referred to as *tunneling*, while the transport of a single electron is referred to as a *tunnel event*. Electrons are considered to tunnel through a tunnel junction strictly one after another.

Rather than calculating for each tunnel junction if a hypothetical charge event results in a reduction of the circuit's energy, we can calculate the critical voltage V_c, which is the voltage threshold needed across the tunnel junction to make a tunnel event through this tunnel junction possible. For calculating the critical voltage of a junction, we assume a tunnel junction with a capacitance of C_j. The remainder of the circuit, as viewed from the tunnel junction's perspective, has an equivalent capacitance of C_e. Given the approach presented in [5], we calculate V_c for the junction as $V_c = \frac{-q_e}{2(C_e+C_j)}$, where $q_e = 1.602 \cdot 10^{-19}\ C$ is the charge of the electron.

Generally speaking, if we define the voltage across a junction as V_j, a tunnel event will occur through this tunnel junction if and only if $|V_j| \geq V_c$. If tunnel events cannot occur in any of the circuit's tunnel junctions, i.e., $|V_j| < V_c$ for all junctions in the circuit, the circuit is in a *stable state*. For our research, we focus on circuits where a limited number of tunnel events may occur, resulting in a stable state. Each stable state determines a new output value resulting from the distribution of charge throughout the circuit.

The tunneling of electrons in a circuit containing tunnel junctions is a stochastic process. This means that the delay cannot be analyzed in the traditional sense. However, the orthodox theory for single electron tunneling (see, for example, [5] for a more extensive introduction) provides means to calculate the average number of tunnel events per second. Assuming that individual tunnel events can be described by a Poisson process and given an acceptable error probability P_{error}, the delay t_d of the tunnel event can be calculated as

$$t_d = \frac{ln(P_{error})q_e R_t}{|V_j - V_c|}, \tag{1}$$

where R_t is the tunnel resistance (usually $\approx 10^5\Omega$).

One of the advantages of the SET technology in general is that SET tunnel junctions can be fabricated in many different ways. In order to illustrate the variety in possible implementation technologies, Fig. 1 presents two possible implementations of the SET inverter. Fig. 1a depicts a SET inverter fabricated in a conventional lithographic technology on silicon [6]. In this case, the tunnel junctions resemble conventional capacitors and consist of small gaps between conducting plates. Fig. 1b, on the other hand, depicts a carbon

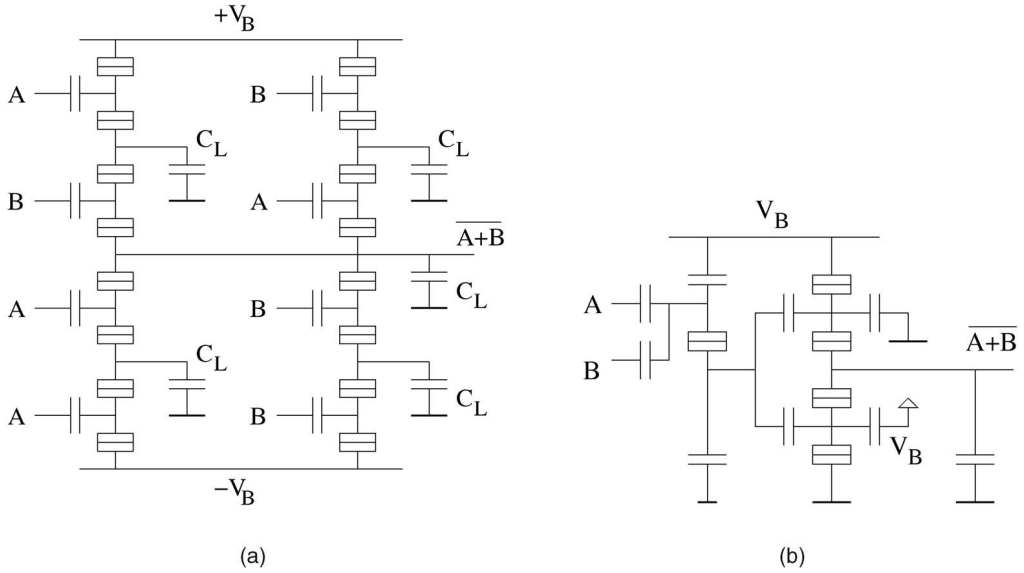


Fig. 2. SET-based NOR gate implementations. (a) CMOS-like design. (b) SEEL TLG design.

nanotube-based implementation [7]. In this case, the tunnel junctions consist of small gaps in a multiwall carbon nanotube.

Besides the switching error probability, there are two fundamental phenomena that may cause errors in SET circuits: thermal tunneling and cotunneling. Given a maximum acceptable switching error probability, we must ensure that the thermal error probability as well as the cotunneling error probability are of the same order of magnitude or less. Thermal tunneling errors are caused by thermal agitation. The thermal error probability can be calculated as $P_{therm} = e^{-\Delta E/K_b T}$, where k_b is Boltzman's constant ($k_b = 1.38 \cdot 10^{-23} J/K$), T is the operating temperature, and ΔE is the change in the energy present in the system as a result of the tunnel event. Note that ΔE scales inversely with the capacitor sizes in the SET circuit. For a multijunction system in which a combination of tunnel events leads to a reduction of the energy present in the entire system, there exists a nonzero probability that those tunnel events occur simultaneously (even if $|V_j| < V_c$ for all individual tunnel junction involved). This phenomenon is commonly referred to as *cotunneling* [8], [9]. The cotunneling error probability can be reduced sufficiently through the addition of strip resistors [10], [11], [12]. Additionally, current experimental SET circuits contain random electrical charges which affect circuit biasing. Such charges are assumed to be the result of trapped charge particles in the tunnel junctions themselves or in the substrate and are anticipated [3] to reduce or even disappear entirely for the nanometer-scale feature size circuits required for room temperature operations.

Thus far, the research on SET-based logic has predominantly focused on two design styles. The CMOS-like design style is based on the SET transistor (see [13] for an early review paper), which consists of two tunnel junctions in series and a capacitor or a resistor attached to the interlaying circuit node. The resulting three terminal structure can be biased such that behaves similarly to a p or an n transistor. Thus, one can convert existing CMOS cell libraries to their SET equivalents. Various SET transistor logic families have been proposed [14], [15], [16], [17]. The common denominator is that these designs attempt to copy CMOS gate structures. Fig. 2a, for example, presents a CMOS-like NOR gate implementation [15]. The main advantage of converting existing cell libraries is the reutilization of existing knowledge and tools. Once a family of Boolean logic gates has been developed in a novel technology such as SET, existing gate-level designs of (larger) components, such as adders, multipliers, etc., can be realized in a straightforward manner. Equally important, existing design tools can be ported at very little cost and effort.

The second design style, generally referred to as single electron logic, is based on encoding the Boolean values directly as single electron charges. One approach in this direction, as first suggested in [18], is based on the physical transport of charge from one gate to another such that Boolean input signals consist of the presence or absence of arriving charge. Another approach, as first suggested in [19], is based on scaling down the charge transport in SET transistor-based structures to a few electrons and confining charge transport within individual gates. When charge transport is scaled down to just one electron, this approach leads to Single Electron Encoded Logic (SEEL) logic in which the Boolean logic values 0 and 1 are encoded as a net charge of 0 and 1 electron charge only [19]. The SEEL paradigm can be applied to construct compact SEEL Threshold Logic Gates [20], [21]. Fig. 2b, for example, presents a SEEL TLG-based NOR gate implementation [21].

As one can observe in Fig. 2, the CMOS-like design style is less area efficient. The CMOS-like NOR gate requires 25 circuit elements, whereas the SEEL threshold gate-based NOR gate requires only 14 circuit elements. Also, the power consumption of the CMOS-like gate is larger than that of the SEEL TLG-based gate as it not only transports a larger amount of charge, but also consumes a static current. Finally, the delay of the CMOS-like NOT gate is on the order of 10 ns [15], while a SEEL TLG-based NOR gate has a delay on the order of 1 ns [21].

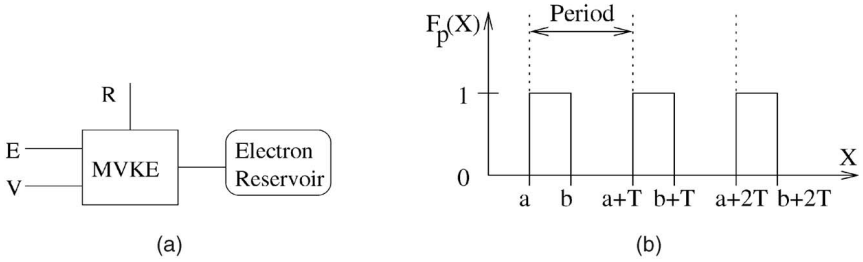


Fig. 3. Building blocks for electron counting-based arithmetic. (a) The $MVke$ block. (b) Period symmetric function $F_p(X)$.

Although the SEEL approach utilizes the SET technology more efficiently than the CMOS-like approach, it still does not use the SET technology's full potential. While SEEL is still based on Boolean variables, the majority of computational and storage logic is intended for multibit variables (e.g., n-bit adders, registers, etc.). Thus, a paradigm that can operate directly on such operands will potentially lead to more effective computation. The next section discusses such an approach in detail.

3 Electron Counting-Based Arithmetic

Given that the SET technology allows one to control the transport of individual electrons, we have the possibility of encoding integer values X directly as a net extra charge Xq_e. Once integer values have been encoded as a number of electrons, we can perform arithmetic operations directly in electron charges. This reveals a broad range of novel computational schemes, which we will generally refer to as electron counting.

In this section, we assume binary encoded n-bit operands, $A = (a_0, a_1, \ldots, a_{n-1})$ and $B = (b_0, b_1, \ldots, b_{n-1})$, and discuss electron counting schemes to compute the result of their addition and multiplication. The basic idea behind the method [22] is first to convert the operands from digital to charge representation, add/subtract them in charge format, and convert the result back to binary digital representation. Before describing the concept in more detail, we briefly discuss the two types of electron counting building blocks which are required for these schemes.

The $MVke$ block depicted in Fig. 3a is an electron counting building block with which a variable number of electrons can be added to or removed from a charge reservoir. Thus, it can be utilized to move electrons within an SET circuit. Typically, a charge reservoir is a circuit node that is capacitively coupled to ground. A charge reservoir with a capacitance C_{cr} containing a charge of $V \times q_e$ is therefore equivalent to a voltage source $U = \frac{V \times q_e}{C_{cr}}$. The $MVke$ block behavior is controlled via two Boolean input signals, R (reset) and E (enable), and it operates as follows: If $R = 0$ and $E = 1$, $V \times k$ electrons are removed from the electron reservoir, where k is a positive integer constant and V is an integer (variable) value. Note that V could either be another charge reservoir containing a charge $V \times q_e$ or an equivalent voltage source. For positive V values, the $MVke$ block is in "add" mode (increasing the charge of the reservoir by removing electrons) while, for negative V values, the $MVke$ block is in "remove" mode (reducing the charge of the reservoir by adding electrons). The $MVke$ block has a dynamic logic behavior. Thus, before a new charge transport can be initiated, it has to be reset, which can be achieved by $R = 1$ and $E = 0$.

A Boolean symmetric function $F_s(x_0, x_1, \ldots, x_{n-1})$ is a Boolean function for which the output depends on the sum of the inputs $X = \sum_{i=0}^{n-1} x_i$. A Periodic Symmetric Function (PSF) $F_p(X)$ is a symmetric function for which $F_p(X) = F_p(X + T)$, where T is the period. Any PSF can be completely characterized by T, the value of its period, and a, b, the values of X corresponding to the first positive transition and the first negative transition, as displayed in Fig. 3b. Efficient implementation of periodic symmetric functions is quite important as many functions involved in computer arithmetic computations, e.g., parity, belong to this class of functions. The PSF block is an electron counting building block that can evaluate a PSF, where it is assumed that the sum of the inputs X is charge encoded and stored in a charge reservoir.

Given these two types of building blocks, we can now discuss electron counting schemes for addition and multiplication. Assuming binary operands, the first step in any electron counting process is to convert a binary integer value X to its discrete analog equivalent Xq_e using a Digital to Analog Converter (DAC) which follows the general organization of the one introduced in [23]. As described earlier, the $MVke$ block in Fig. 6a can be utilized to add/remove a number of electrons to/from a charge reservoir. When multiple such $MVke$ blocks operate in parallel on the same charge reservoir, electrons can be added to or removed from the reservoir in parallel. More specifically, to convert an operand $X = (x_0, x_1, \ldots, x_{n-1})$, each bit x_i, $i = 0, 1, \ldots, n-1$ is connected to the E input of an $MVke$ block that has the V input hardwired to a bias potential that induces a $V \times k$ value equal to 2^i. Therefore, the operand X can be encoded as $\sum_{i=0}^{n-1} x_i 2^i q_e$ at the cost of n $MVke$ blocks in "add" mode. Thus, this DAC scheme has an $O(n)$ asymptotic complexity in terms of the number of required building blocks.

Given the $MVke$-DAC encoding scheme described above, the addition of two n-bit operands, A and B, can be embedded in the conversion process if the operands are converted into charge format via a total of $2n$ $MVke$ blocks in "add" mode that share the same charge reservoir. Once the result corresponding to the addition is available in the charge reservoir as a charge Yq_e, where $Y = A + B$, we need to convert this result back to a digital format in order to finalize the computation process. To achieve this, an Analog to Digital Conversion (ADC) process is required. In the following, we describe an ADC circuit based on the PSF block.

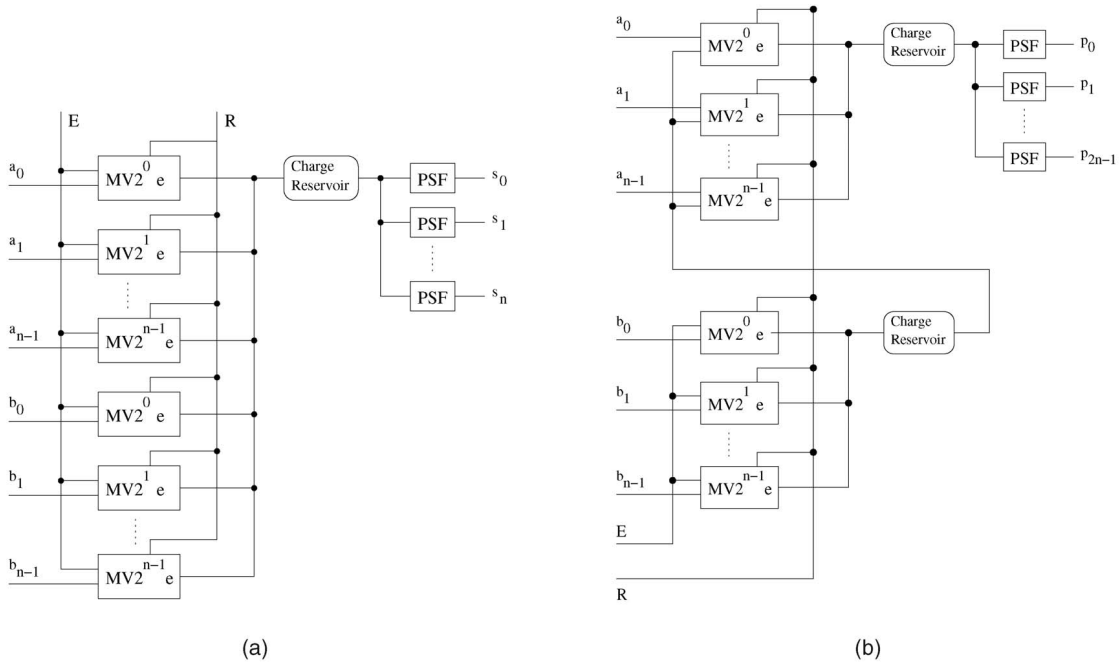


Fig. 4. Electron counting addition and multiplication schemes. (a) n-bit addition circuit. (b) n-bit multiplication circuit.

If N is the maximum number of extra electrons that can be removed from the result charge reservoir, $m = 1 + [\log N]$ bits are required to represent this value in binary format. Then, following the base 2 counting rules, any ADC output bit s_i, $i = 0, 1, \ldots, [\log N]$ is equal to 1 inside an interval that includes 2^i consecutive integers, every 2^{i+1} integers, and 0 otherwise. Thus, each bit s_i can be described by a periodic symmetric function with period 2^{i+1}. As a consequence of this property, each output bit s_i can be computed by a PSF block that had been adjusted in order to have a transfer function that copies the periodic symmetric function required for the bit position i. Thus, we can implement an m-bit ADC using m PSF blocks (the PSF applied at bit position i is tuned to exhibit the periodic transfer function corresponding to that s_i bit) assuming as input the same charge reservoir. Given that we are addressing the particular case of n-bit operand addition such that $m = n + 1$, the cost of the required ADC circuit is on the order of $O(n)$.

Summarizing, the electron counting-based addition of two n-bit operands can be implemented with a depth-2 SET network composed out of $3n + 1$ electron counting building blocks, then with an $O(n)$ asymptotic complexity measured in terms of building blocks. The overall organization of the circuit is depicted in Fig. 4. We note here that, in the figure, the value k of the $MVke$ blocks has been drawn inside the blocks to suggest that it was implemented by properly adjusting the corresponding circuit parameter(s), while all inputs V have been fixed to the equivalent of a charge reservoir with $1q_e$ charge.

As our main goal was to demonstrate a new paradigm for addition, we assumed, for simplicity, that the operands are unsigned numbers. However, the scheme in Fig. 4a can be easily extended to operate with signed numbers. Given that the sign information is actually present in the charge reservoir, a simple threshold logic gate evaluating $sgn\{ChargeReservoir\}$ can produce the correct sign of the results. Thus, if we assume, for example, $(n + 1)$-bit sign-magnitude operands A and B, the sum $A + B$ can evaluated by the circuit in Fig. 4a if the sign bits of the operands are used as control signals for the V input of the $MVke$ blocks as follows: For all the $MVke$ blocks that process an operand, $V = -1$ if the sign bit is 1 and $V = 1$ otherwise. To evaluate $A - B$, we just have to invert the sign bit of B before computing the V values for the B operand. The magnitude of the result is correctly evaluated by the PSF blocks and, for the sign bit of the result, we just have to augment the circuit in Fig. 4a with one threshold gate.

Even though the proposed addition scheme is primarily meant for addition/subtraction, it has a broader scope. Some of the alternative utilizations include n-bit subtraction, n-bit parity functions, multioperand addition, and $n|\log n$ counters, as described in [22].

We next discuss an electron counting multiplication scheme that follows, to some extent, the paradigm we introduced for addition. Assume we have the input operands A and B and we want to compute $P = A \times B$. As indicated in [22], a straightforward application of the electron counting principle to the multiplication produces a depth-3 network with an overall asymptotic complexity measured in terms of circuit elements in the order of $O(n^2)$. A more effective implementation is also possible if one makes use of the ability to transport a variable number of electrons to/from a charge reservoir exhibited by the $MVke$ structure depicted in Fig. 3a.

The basic idea behind the scheme is again to build up a charge $P \times q_e$ in a charge reservoir and to utilize an ADC structure to obtain the binary representation of the product P. The general organization of the proposed multiplication circuit is depicted in Fig. 4b. Again, the value k of the $MVke$ blocks has been drawn inside the blocks themselves to suggest that that k value was implemented inside the block by

properly adjusting the corresponding circuit parameter(s). The scheme is utilizing a clock for synchronization purposes[2] and the computation process can be described as follows: First, on the positive clock value, a charge corresponding to the value of the B operand, i.e., $\sum_{i=0}^{n-1} b_i 2^i q_e$, is built up in the corresponding charge reservoir. This is achieved with n $MVke$ blocks, each of them assuming as inputs the b_i bits and having the V input hardwired to the equivalent of a charge reservoir with $1q_e$ charge, such that $V \times k = 2^i$. Second, on the negative clock value, a charge of $A \times B \times q_e$ is built up in the other charge reservoir. This is achieved with n $MVke$ blocks, assuming as inputs the a_i bits and the analog value present on the charge reservoir processed in the previous computation step. As each $MVke$ block in this stage contributes $a_i \times 2^i \times B$ electrons, a final charge of $\sum_{i=0}^{n-1} a_i 2^i \times Bq_e$, i.e., $A \times B \times q_e$, is present in the output charge reservoir when the second step is completed. Last, the value on the output charge reservoir is converted to digital with $2n-1$ PSF blocks.

This scheme still implies a depth-3 network, but requires $2n$ $MVke$ blocks and $2n-1$ PSF blocks, thus the overall asymptotic complexity is reduced to $O(n)$.

4 Electron Counting—Practical Considerations

In the previous discussion about addition and multiplication via the electron counting paradigm, we were mainly interested in establishing the asymptotic bounds for the delay and area of adders and multipliers potentially implemented in this paradigm. While these bounds indicate that electron counting-based arithmetic circuits have the potential to outperform Boolean and Threshold logic-based circuits, they are mainly of theoretical interest. Whether or not this holds in practical implementations very much depends on a number of issues that are fabrication technology dependent, e.g., the maximum number of electrons that can be accurately manipulated via an $MVke$ block, the dependence of the block's delay on the capacitance of the charge reservoir, the maximum value in an electron reservoir that can be accuracy handled by a PSF block, etc. Additionally, the computation via charge manipulation requires the transport of at most 2^{n+1} electrons for n-bit addition and at most 2^{2n} electrons for n-bit multiplication. This also limits the operand width and affects the calculation delay as we expect that the more electrons have to tunnel, the larger the delay.

While, for a mature technology like CMOS, such things can be easily characterized, this does not hold true for an emerging technology like SET. There is very little we can say about the actual values of the device parameters, thus, by implication, about the previously mentioned issues. Regardless of this, we can with certainty assume that, for a given SET fabrication technology, the maximum number of electrons that can be accommodated in a charge reservoir is limited by a certain value and that this value limits the amount of charge that can be manipulated by the $MVke$ blocks in the circuit. The constant depth addition and multiplication designs have to be changed in order to deal with such limitations. Thus, such a restriction has a direct implication on the delay of electron counting-based designs. In the remainder of this section, we discuss a number of mechanisms that one can use in order to cope with such practical restrictions. It is not our intention to exhaustively cover this issue (future research is still required in order to better understand all the involved phenomena); we only attempt to demonstrate that interesting solutions exist and evaluate the implication of such restrictions on performance and area.

Let us assume the n-bit addition case and that at most 2^r-1 electrons can be accurately controlled by an $MVke$ block. This implies that $V \times k$ can span between 0 and 2^r-1. To be able to construct an electron counting-based implementation under this restriction, we have to partition the operands in r-bit groups, which is equivalent to a radix-2^r operation. This produces, in the general case, $\lceil \frac{n}{r} \rceil$ groups, but, for simplicity of notation, we can assume with no lost of generality that r is a divisor of n and neglect the ceiling operator.

The proposed addition scheme is depicted in Fig. 5 and it operates as follows: For each and every digit position j, $j = 1, 2, \ldots, \frac{n}{r}$, we can use the scheme described in Section 3 and convert in charge, in a charge reservoir CR^j, the sum of the two digits $T^j = A^j + B^j$. As these values can be larger than 2^r-1, a carry into the next digit might be produced. To evaluate if such a carry has to be considered into the next position, we can use a conditional $MVke$ block ($CM1e$) that assumes as input CR^j and removes one electron from the charge reservoir CR^{j+1} in the case that $sgn\{CR^j - 2^r\} = 1$, i.e., a carry was produced, and leaves CR^{j+1} unchanged otherwise. The process continues in this way until the last charge reservoir $CR^{\frac{n}{r}}$ is updated. At this point, the carry propagation between the digits is completed and the charge to digital conversion can be performed in parallel for all the charge reservoirs by enabling all the PSF blocks. This new scheme requires $2n$ $MVke$ blocks, $n+1$ PSF blocks, and $\frac{n}{r}$ $CM1e$ blocks. When considering the delay of this scheme, the critical path includes an $MVke$ block, $\frac{n}{r}$ $CM1e$ blocks, and a PSF block. Thus, when we compare this scheme with the one in Section 3, the extra delay is on the order of $\frac{n}{r}$ and, as expected, the higher the radix one can assume, the lower the delay penalty. We note, however, that, given that the $MVke$ blocks and PSF blocks can become slower when k is larger than a certain value, the highest radix possible for a given technology might not necessarily mean the fastest scheme. Moreover, we note here that the $CM1e$ blocks are faster than the normal $MVke$ blocks as, for them, at most one electron is removed from the output charge reservoir.

The extra delay due to the carry propagation between digit positions can be further reduced if a carry lookahead technique is utilized. In this way, based on the values in the charge reservoirs, we can compute the generate, digit propagate, and carry signals for all digit positions following, for example, the method in [24]. In this way, the extra delay is brought in the order of $\log \frac{n}{r}$ at the expanse of some additional TL gates.

A similar technique can be applied to the multiplication when the fabrication technology does not allow for the direct utilization of the electron counting multiplication scheme discussed in Section 3. Under the same assumption that at most 2^r-1 electrons can be accurately controlled by the $MVke$ blocks, a possible method to implement multiplication

2. We assume here a level triggered behavior, but the scheme can work with edge triggered policy as well.

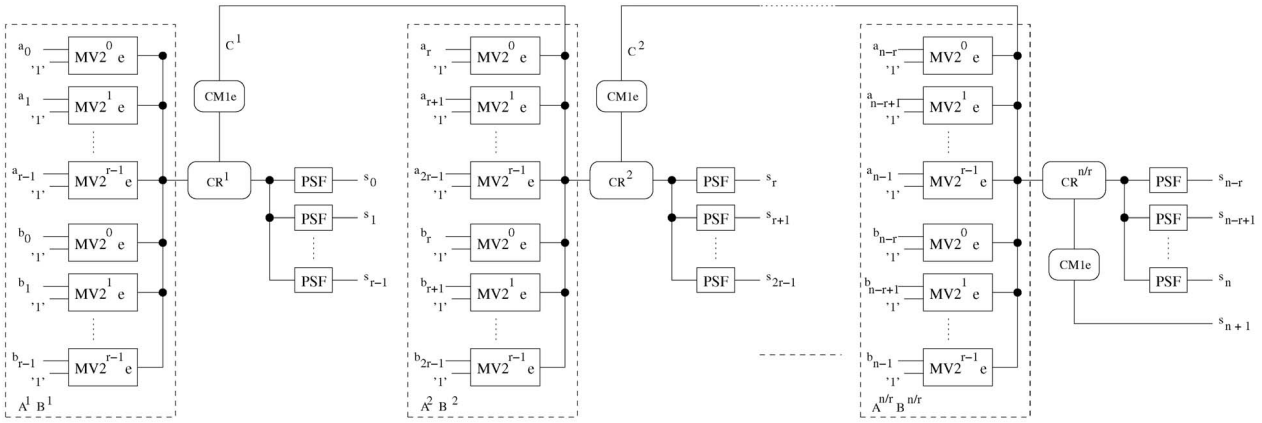


Fig. 5. Organization of high radix n-bit addition circuit.

can be sketched as follows: Again, we can operate in high radix, but, in this case (in order to keep the maximum number of electrons within boundaries), we have to assume that the input operands are in radix-$2^{\sqrt{r}}$, thus we initially deal with $\sqrt{r}$-bit digits. First, we can compute the partial products by applying the depth-3 multiplication scheme in Fig. 4b at the digit level. We note here that, due to the fact that the next computation steps are performed directly on digits kept in charge reservoirs, the partial products do not have to be reconverted in binary after the multiplication. This means that the multiplication we use for the production of the radix-2^r partial products is depth-2 as the final conversion is not required. This first step requires $\frac{n^2}{r}$ such depth-2 multipliers. Next, we reduce the partial product matrix that contains r-bit digits to two rows. This can be achieved with various types of counters that operate on and produce radix-2^r digits kept in charge reservoirs. Such counters can be built with conditional $MVke$ blocks that can now remove a variable number of electrons from the output charge reservoir. A conditional $MVke$ block can be built by augmenting the standard $MVke$ block with a threshold gate. At the end of the reduction process, the two available rows are formed by charge reservoirs containing radix-2^r digits, thus an adder based on the scheme in Fig. 5 can be utilized to complete the calculation. Again, this adder is less complex as the inputs are already converted in charge. Asymptotically speaking, the overall multiplication delay might be on the order of $O(\log \frac{n}{r})$. For the time being, we cannot make more accurate delay evaluations as it very much depends on the type of counters one can implement under the constraints we assumed and this issue is subject to future research.

Thus far, we have discussed potential schemes for addition and multiplication. In order to evaluate implementations of the proposed electron counting addition and multiplication schemes, the next section proposes possible implementations of the required building blocks.

5 Possible Implementations of Electron Counting Building Blocks

In Section 3, we discussed several approaches for computing arithmetic functions, e.g., addition and multiplication, via the controlled transport of individual electrons. These proposals are based on the $MVke$ and PSF building blocks, whose functional behavior was also introduced in Section 3. In this section, we propose possible SET-based implementations of the these two building blocks. We discuss their operation principles and demonstrate the behavior of the designs by means of simulation.

5.1 Implementing the MVke Building Block

The SET transistor can be utilized to control the transport of charge from its source to its drain terminal. For any applied gate voltage V_g, there exists a minimum drain-source voltage V_{ds} that results in a nonzero current i_d. However, if the source and/or the drain terminal are capacitively coupled to their environment, the transported charge will (gradually) reduce V_{ds} until further charge transport is no longer possible. This principle forms the basis for the proposed $MVke$ block, as depicted in Fig. 6a. The Boolean inputs E (enable) and R (reset) are control signals and the input V is a discrete analog input representing an integer value.

Before discussing the operation principle of the proposed implementation we have to make certain assumptions related to the electrical representation of the logic "0" and logic "1" values. In the remainder of this paper, if not specified otherwise, we assume that Boolean input/output signals correspond with the following voltages: logic "0" = 0 Volt, logic "1" = $q_e/10C$ Volt, where C acts as a unit for capacitance. If, for example, we choose $C = 10^{-18}F$, we find logic "1" = 16 mV.

The proposed implementation of the $MVke$ block operates as follows: Assume that, initially, all inputs are 0 and the circuit is in a neutral charge configuration, i.e., the net charges present on circuit nodes t and i as well as the charge present in the charge reservoir are 0. When the circuit is enabled ($E =$ "1"), the voltage across junction C_{j1} becomes close to its critical voltage V_c. If, simultaneously, the input V assumes a value larger than 0, the voltage across junction C_{j1} becomes larger than its critical voltage and charge transport occurs. Note that this charge transport consists of paired tunnel events: A tunnel event in junction C_{j1} (transporting one electron from node i to node t) is followed by a tunnel event in junction C_{j2} (transporting one electron from the charge reservoir to node i). The net effect of each paired tunnel event is $+1q_e$ charge present in the charge reservoir and $-1q_e$ charge present on node t.

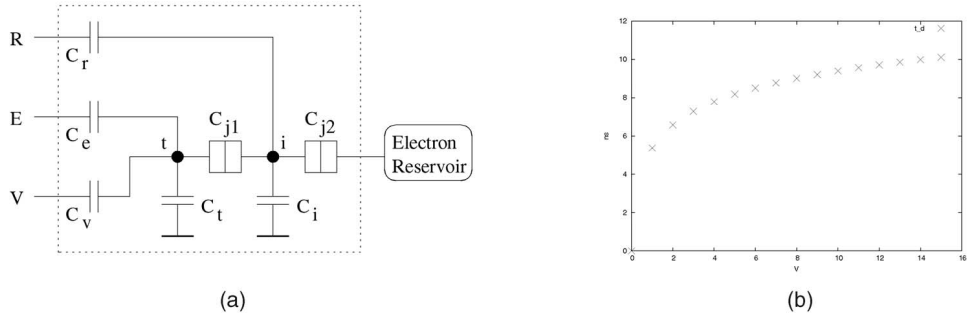


Fig. 6. Implementation details of the $MVke$ block. (a) Circuit diagram. (b) Delay as a function of V.

Eventually, the voltage resulting from the increasing negative charge present on node t will cancel the effect of the input signal V and further charge transport will cease. Note that the circuit can be designed such that both V and E can now be set to "0" without losing the charge trapped on node t. This has the added advantage that any effect of V and E on the output voltage (due to capacitive voltage division) is removed entirely. After the result produced by the $MVke$ block has been processed by the next circuit element, the $MVke$ block can be reset ($R =$ "1") while $E = V = 0$, reversing the process described above. In this way, the $MVke$ block is brought back to the neutral charge configuration and made ready for the next evaluation phase. We note here that one simple way to make $E =$ "0" and $V = 0$ during the reset phase is to NOR them with R. In this way, they are always 0 when $R =$ "1" and R can be seen as the only control signal.

In order to implement an instance of the $MVke$ block described above, we assume that the capacitance C_{cr} of the charge reservoir is much larger than the capacitance of C_{j2} ($C_{cr} >> C_{j2}$). We also assume that $C_{\Sigma t} >> C_{j1}$, where $C_{\Sigma t} = C_e + C_v + C_t$. As a result, we found, for both junctions, that $V_c = e/2C_{\Sigma i}$, where $C_{\Sigma i} = C_{j1} + C_{j2} + C_r + C_i$. Choosing $C_{\Sigma i} = 10C$ and $C = 10^{-18}F$, we find $V_c \approx 8mV$ for both junctions. However, given these assumptions, we calculated that the actual critical voltage is somewhat higher. We therefore chose the circuit parameters such that the voltage across junction C_{j1} is 8 mV when E = logic "1" = 16 mV. In order to decouple the input from the output, as well as to maximize the contributions of the input voltages E and V to the voltage across junction C_{j1}, we choose $C_{j1} = C_{j2} = 0.5C$. As a result, we calculated that the following capacitor ratios are required: $C_e/C_{\Sigma t} = 5/9.5$ and $C_r/C_{\Sigma i} = 5/10$. The integer constant k of the $MVke$ block is determined by the capacitor C_v as follows: Assume $V = 1$ corresponds to a voltage $q_e/\alpha C$ and that k electrons have been transported to node t. In order for the contribution of the input V and of the transported charge $k \times q_e$ to the voltage of node t to cancel each other, we find that $C_v/\alpha C = k$. Assuming, for example, $\alpha = 100$, we find that $k = 3$ corresponds to $C_v = 300C$.

To demonstrate the correct behavior of the proposed $MVke$ block, we have simulated an instance of the circuit. We utilized $C = 1aF$, $\alpha = 100$, $k = 3$, which resulted in the following circuit parameters (corresponding to the previous discussions): $C_{j1} = C_{j2} = 0.5aF$, $C_r = 5aF$, $C_i = 4aF$, $C_e = 500aF$, $C_v = 300aF$, $C_t = 150aF$, logic 1 = 16 mV. The charge reservoir is implemented by a capacitor $C_{cr} = 10^{-14}F$. The simulation results are presented in Fig. 7. In the figure, the top three bars represent the inputs R, E, and V, while the bottom two bars represents the charge present in the charge reservoir and the voltage across C_{cr} (the capacitor implementing the charge reservoir). As can be observed in the figure, the $MVke$ block transports $V \times 3q_e$ charge to the charge reservoir as it should. Given that $C_{cr} = 10^{-14}F$, every $3q_e$ charge added corresponds to an increase of 0.048 mV. Maintaining signal strength therefore requires an amplification of factor 100. As one can observe, the charge reservoir maintains its value when $E = R = V = 0$ Volt. Finally, no charge transport occurs while $E = 0$ as it can be observed that, for the last input value, $V = 8$ mV.

Given the above parameters, we can estimate the delay of the $MVke$ block as a function of the input value V. Charge transportation through a tunnel is assumed to be sequential in nature. Thus, $V = 2$ (and $k = 3$), for example, results in six sequential *combined* tunnel events. By a combined tunnel event, we refer to an initial tunnel event in junction C_{j1}, followed by a tunnel event in junction C_{j2}. As the delay of the second event can be neglected when compared with the delay of the first event, the delay estimates are based on the first event only. Each tunnel event in junction C_{j1} reduces the voltage V_j across this tunnel junction. Given (1), this implies that each consecutive tunnel event has a larger delay than the previous one. The

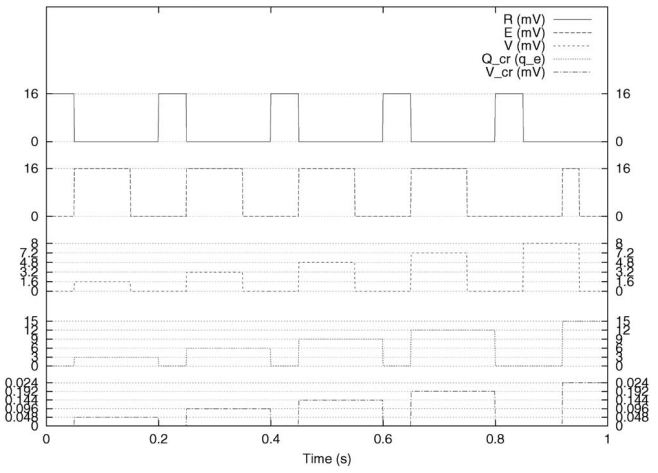


Fig. 7. $MVke$ block simulation results.

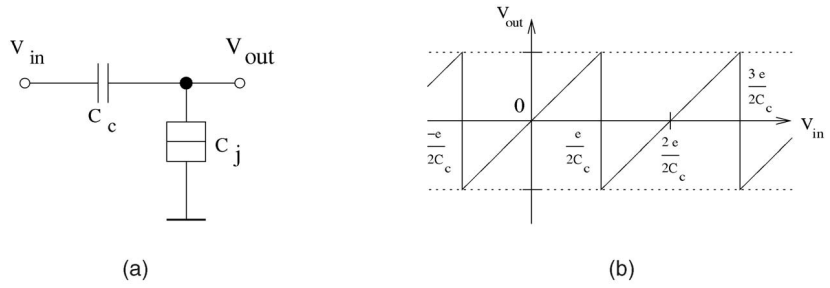


Fig. 8. SET electron trap. (a) Circuit. (b) Transfer function.

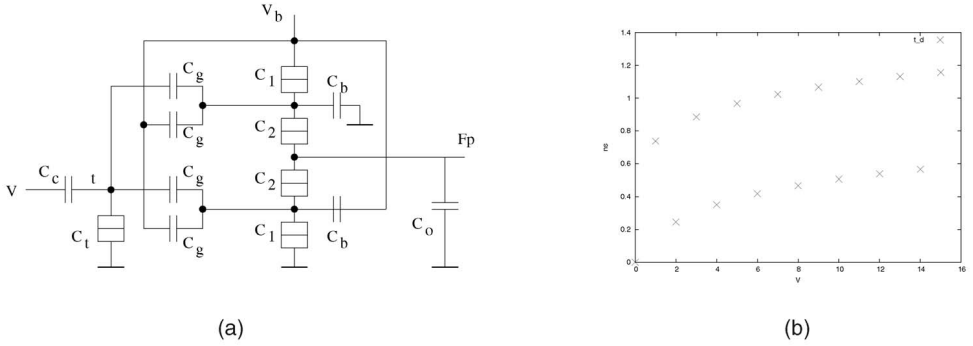


Fig. 9. Implementation details of the PSF block. (a) Circuit diagram. (b) Delay as a function of V.

total delay can be calculated by summarizing each individual delay, which, for $V > 0$, results in:

$$t_d = \Sigma_{i=1}^{V \cdot k} \frac{0.295}{1.6 \cdot \frac{3}{9.5} \cdot \frac{2i-1}{2k}} \; ns. \tag{2}$$

In order to visualize the above result, we plotted the delay t_d as a function of the input value V. The result is depicted in Fig. 6b. From this, it can be observed that, although the delay depends on the number of electrons which are transported through the tunnel junctions, the increase in delay is asymptotically bound. As we can observe in Fig. 6b, for the circuit parameters we assumed, the delay varies between 5 and 10 ns, which means that the maximum delay is about the same as the typical delay of a CMOS-like gate, but $10\times$ larger than the one of a SEEL gate. The area of the $MVke$ block can be estimated in a straightforward manner by counting the required number of circuit elements. Excluding the charge reservoir capacitor, we find that the area costs seven circuit elements.

5.2 Implementing the PSF Building Block

The electron trap, consisting simply of a capacitor in series with a tunnel junction, as depicted in Fig. 8a, displays a periodic behavior. Thus, it can serve as the basis for an implementation of the PSF block. The operating principle of the electron trap is as follows: Assume that the input voltage V_{in} starts at 0 Volt such that the net charge present on the output node is 0. When V_{in} is gradually increased, the output voltage will also rise due to voltage division. Eventually, the voltage across the tunnel junction will reach its critical value V_c and an electron will tunnel. As a result of this tunnel event, the voltage present at the electron trap output drops to a negative value. When V_{in} further increases, the output voltage again gradually increases until it passes the critical value, after which the process repeats itself. The transfer function of the electron trap, as depicted in Fig. 8b, is therefore periodic in nature.

The periodic signal of the electron trap and the literal function of the SET inverter serve as the basis for our proposed implementation of the PSF, which is depicted in Fig. 9a. The circuit operates as follows: The SET inverter behaves as a literal gate and transforms its input signal (within a limited range) to either logic 0 or logic 1. The inverter is modified such that it has two inputs. One of its inputs is attached to a bias voltage V_b. The bias voltage is set such that the inverter is close to its switching point. The output of the electron trap serves as the second input to the modified inverter block. Given that the PSF block is intended as a building block for Analog to Digital Conversion (ADC), we are solely interested in periodic symmetric functions (see Fig. 3b) in which $b = 2a$ and $T = 2a$. We therefore bias the inverter in order to obtain the following behavior: If the output of the electron box is negative, the inverter interprets the combined input of the electron trap and the bias voltage as logic 0 and its output becomes 1. Likewise, if the output of the electron trap is positive, the inverter interprets the combined input of the electron trap and the bias voltage as logic 1 and its output becomes 0.

When implementing instances of the proposed PSF block, the following is assumed: The input V of the PSF block is a discrete analog input representing an integer value. Assume that $V = 1$ corresponds to a voltage $q_e/\alpha C$ and that we implement the periodic symmetric function characterized by $a = q_e/\alpha C$, $b = 2q_e/\alpha C$, $T = 2q_e/\alpha C$ (see Fig. 3b). Given the operating principle of the proposed implementation of the PSF block and the general transfer function of the electron trap (see Fig. 8b), we can implement this PSF block by mapping $a = q_e/\alpha C$ to $V = q_e/2C_t$. Assuming $\alpha = 100$, we find $C_c = 50C$. The maximum amplitude of the electron trap is determined by the total

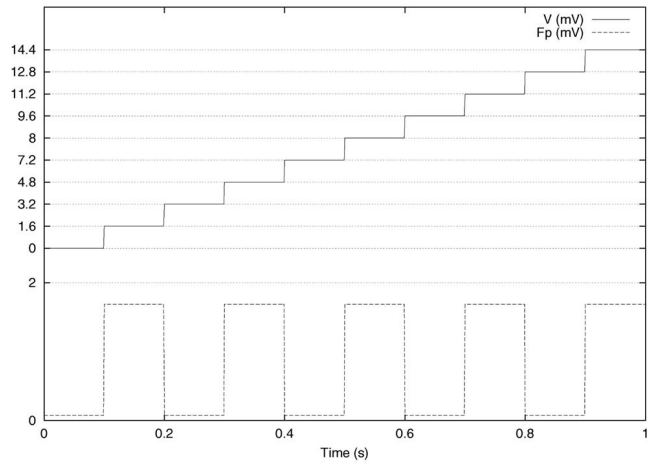


Fig. 10. PSF block simulation results.

capacitance attached to node t. Choosing $C_{\Sigma t} = \alpha C$ results in a maximum signal amplitude $V_{max} = q_e/\alpha C$.

In order to simplify the remainder of the design process, we assume that the bias voltage V_b and the output of the electron trap have equally weighted contributions to the modified inverter, i.e., equal sized capacitors C_g. Additionally, we assume $\alpha = 100$, resulting in the following set of circuit parameters for the inverter: $C_1 = 1C$, $C_2 = 5C$, $C_b = 42.5C$, $C_o = 90C$, $V_b = q_e/100C$. Note that these parameter ratios have previously been verified (see, for example, [20]) and scale linearly with α (such that, for example, $\alpha = 10$ corresponds with $C_2 = 0.5C$).

Assuming $C = 10^{-18}F$, we verified the resulting design by means of simulation (SIMON) using the following circuit parameters: $C_c = 50aF$, $C_j = 45aF$, $C_g = 2.5aF$, $C_1 = 1aF$, $C_2 = 5aF$, $C_b = 42.5aF$, $C_o = 90aF$, $V_b = 1.61V$, $\alpha = 100$ ($V = 1 = 1.6$ mV). The simulation results are depicted in Fig. 10. In the figure, the top represents the analog input V while the bottom bar represents the binary output F_p. As can be observed, the PSF block is performing the expected computation.

The delay calculations of the PSF block are somewhat complicated as they cannot be described by a single equation. This can be explained as follows: Both when $V = 1$ and when $V = 2$, the electron trap transports 1 electron. However, when $V = 2$, the voltage V_j across the electron trap's tunnel junction C_j is larger. Thus, the difference between V_j and the critical voltage V_c is larger, which implies (see (1)) that the delay for $V = 2$ is less than the delay for $V = 1$. This holds for all consecutive pairs such as $V = 3$ and $V = 4$, $V = 5$ and $V = 6$, etc. Given the parameters described above and (1), we calculated the delay of the PSF block for various values of V. The results are depicted in Fig. 9b. It can be observed that, for increasing values of V, the delay of odd V values is asymptotically bound to approximately double the delay of even V values. As we can observe in the figure, for the circuit parameters we assumed, the delay varies between 0.2 and 1.2 ns, which means that the maximum delay is about the same with the typical delay of a SEEL gate.

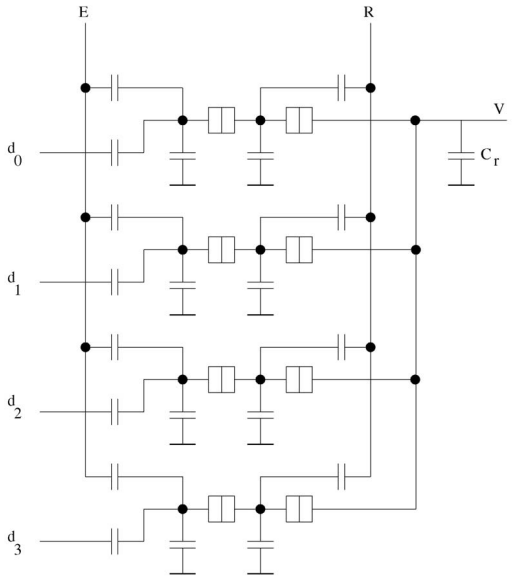


Fig. 11. 4-bit digital to analog converter.

6 Electron Counting Circuit Examples

In the previous section, we proposed SET-based implementations of the $MVke$ and PSF blocks. In this section, we present a number of electron counting-based circuit examples. As DAC and ADC are basic steps within the electron counting arithmetic paradigm, we first present such circuits. Subsequently, we present electron counting circuits for addition and multiplications. All the presented circuits are verified via simulation.

6.1 4-Bit Digital to Analog Converter

To convert a 4-bit operand $X = (x_0, x_1, x_2, x_3)$, each bit x_i, $i = 0, 1, 2, 3$, is connected to the E input of an $MVke$ block that has the V input hardwired to a bias potential that induces a $V \times k$ value equal to 2^i. Therefore, the operand X can be encoded as $\sum_{i=0}^{3} x_i 2^i q_e$ at the cost of four $MVke$ blocks with $k_0 = 1, k_1 = 2, k_2 = 4, k_3 = 8$.

To implement an instance of the 4-bit DAC, we assume that, for the binary inputs, logic "1" = 16 mV and logic "0" = 0 mV and that the conversion result is stored in a charge reservoir with the capacitance $C_{cr} = 10^{-14}F$. For each $MVke$ block, we have to compute the C_v value corresponding to the desired k. As in the $MVke$ example, $V = 1$ corresponds to 1.6 mV the input capacitors realizing the weighing (constant k) have to be reduced by a factor of 10. This results in C_v being equal to $10, 20, 40, 80$ aF for $k = 1, 2, 4, 8$, respectively. All the other circuit parameters remain as in the $MVke$ block example presented in Section 5.1. The 4-bit DAC circuit is presented in Fig. 11 and the corresponding simulation results in Fig. 12. Various 4-bit numbers are considered as input and, as the figure indicates, the proposed circuit performs the correct conversions. Additionally, one can observe that, when $E = 0$, this is the case for the last evaluated input pattern, no conversion is performed and the charge reservoir remains in the neutral state.

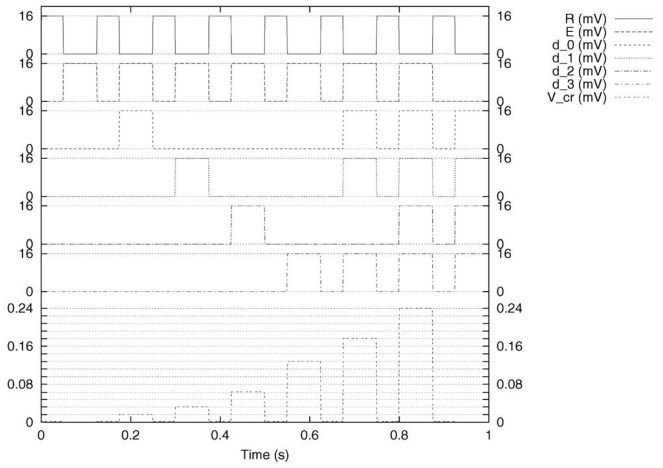


Fig. 12. 4-bit digital to analog converter—simulation results.

6.2 5-Bit Analog to Digital Converter

The next circuit we consider is a 5-bit analog to digital converter (ADC). For such a circuit, the input V is assumed to be the charge encoded on a 10^{-16}F capacitor and the outputs are five binary signals d_0, d_1, d_2, d_3, d_4. To implement the ADC, we need five PSF blocks designed in such a way that each block can evaluate an output signal d_i. For all the $PSFs$, $b = 2a$ and $T = 2a$ and $a_i = 2^i$ for $d_i, i = 0, 1, 2, 3, 4$. To instantiate the five PSF blocks, we have to compute the circuit parameters corresponding to the function they have to evaluate.

We chose $C_t = 20C$ for all the PSF instances. This choice provides enough signal swing on the electron trap output to register small variations, but also not too much maximum amplitude as the inverter has a limited input voltage range for which it can operate correctly. The inverter with the parameters suggested in Section 5.2 is not balanced quite properly. The boundary between inputs considered 0 and 1 shifts, depending on what the last one was. So, for example, a 0 to 1 transition switches the output at $V_{in} = 0.45 * V_{high}$, while a 1 to 0 transition switches at $0.55 * V_{high}$. When the output signal of the electron box changes in very small steps, this might ruin the desired behavior. Experiments revealed that $c_1 = 5C$ (instead of 1C) significantly improves the balance. A slight increase of the V_b to 1.61 mV (instead of the usual 1.6 mV) can also alleviate the situation.

The method for calculating the C_c value is based on some estimates. By applying the estimation method, we found out that the C_c values of 50 aF, 25 aF, 12.5 aF, 6.2 aF, and 3.05 aF, are required for the PSF block calculating d_0, d_1, d_2, d_3, d_4, respectively.

The 5-bit ADC circuit is presented in Fig. 13 and the corresponding simulation results in Fig. 14. As one can observe in Fig. 14, values between 0 and 31 are assumed as inputs and the 5-bit ADC produces the correct results.

6.3 4-Bit Addition

A 4-bit adder structure can be instantiated as a particular case of the addition structure discussed in Section 3. Simply speaking, such a structure can be built with two 4-bit DACs, one $MVke$ block to handle the carry in, a charge reservoir, and one 5-bit ADC. The 4-bit adder circuit is presented in Fig. 15.

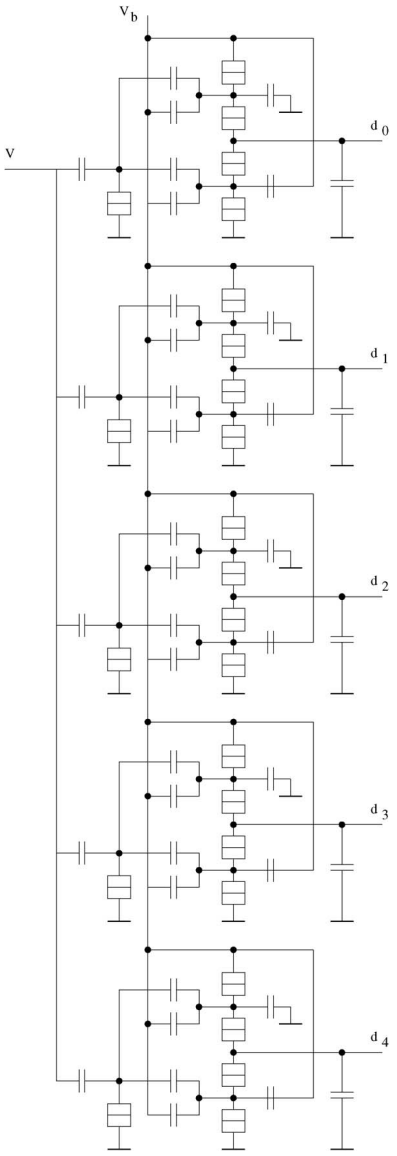


Fig. 13. 5-bit analog to digital converter.

The DAC blocks are identical with the one presented in Section 6.1 and the PSF block is the same with the one described in Section 6.2. The $MVke$ block that processes the C_{in} is identical with the one processing a_0 or b_0.

As the charge reservoir (as in all examples) consists of a 10^{-14} F capacitance, it produces a voltage of 0.016 mV per electron. Due to the fact that the PSF blocks expect an input V such that $V = 1$ corresponds with 1.6 mV, we have to utilize an operational amplifier with 100x amplification factor. Such an amplifier can be built with FET-SET technology and, in principle, SET circuits that include Opamps should not constitute a problem for SIMON simulations. It seems, however, that the utilization of Opamps creates random effects in SIMON simulations. SIMON supports ideal circuit components, so what happens at the output of the Opamp should have no effect at the input side. However, if the Opamp output side has SET circuitry, all kinds of "random" effects start occurring. To solve this problem, we utilized a simulation strategy that requires the partition of the circuit in stages that end with an electron reservoir and an Opamp. In this way, we can

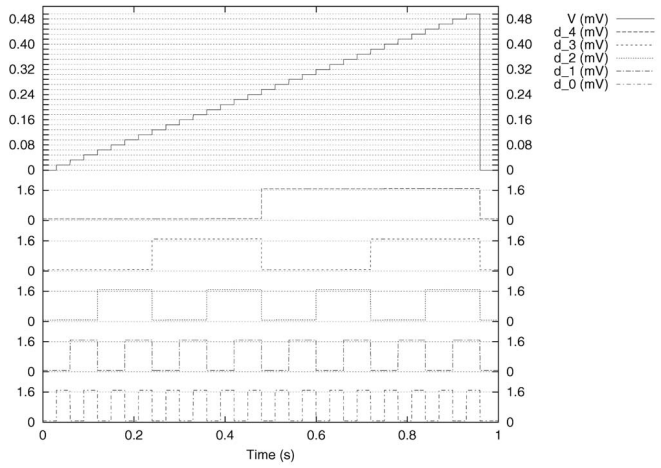


Fig. 14. 5-bit analog to digital converter—simulation results.

first simulate the first stage of the adder and store the simulation data at the Opamp's output. Subsequently, we simulate the next stage, which is now driven by a voltage source instead of an Opamp. By copying the recorded Opamp output values to the SIMON voltage source description such that the voltage source produces exactly the same voltages as the Opamp, we create the appropriate simulation stimuli for the second stage. In this way, the 4-bit adder was simulated in two steps and we avoided the random effects created in SIMON simulations by Opamps followed by SET circuits. The simulation results are presented in Fig. 16 and one can easily observe that the adder functions correctly.

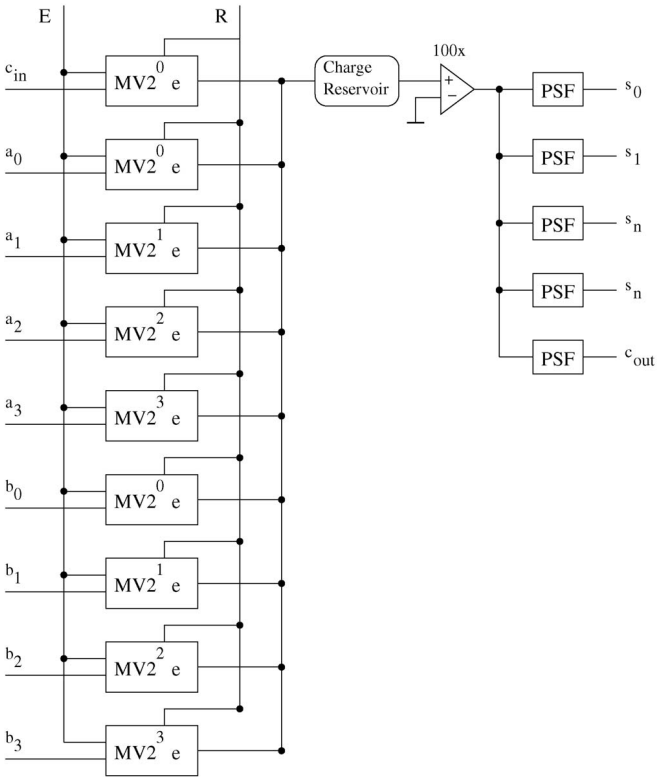


Fig. 15. 4-bit adder.

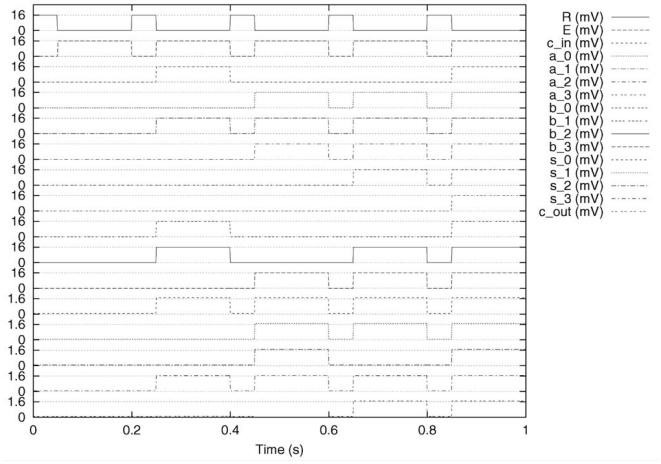


Fig. 16. 4-bit addition—simulation results.

6.4 3-Bit Multiplication

A 3-bit multiplier structure can be instantiated as a particular case of the multiplication structure discussed in Section 3 and it is presented in Fig. 17.

The first stage (the one processing the b_i inputs) is a 3-bit DAC structure that follows the same organization and requires the same parameters as the one required for the first three inputs of the 4-bit DAC. To connect its output at the second stage, an amplification of $100\times$ is required and this can be achieved with an Opamp.

In the second stage, the a_i inputs are attached to the E input of the $MVke$ blocks. Since $E =$ "1" has the same voltage as $a_i =$ "1," no changes are required. The V inputs of the second stage don't have $V = 1$ corresponding with 1.6 mV as in the $MVke$ block example. Thus, for $k = 1, 2, 4$, we have to use $C_v = 100, 200, 400$ aF. To connect the second stage to the third one, a 100x amplification is again required. The last stage is a 6-bit ADC structure that can be obtained by the augmentation of the 5-bit ADC with a PSF block with the C_c value of 1.5aF.

To simulate the multiplier, we utilized the same methodology as for the adder, but now the simulation requires three steps. The simulation results are presented in Fig. 18 and one can easily observe that the multiplier functions correctly.

7 Conclusions

Single Electron Tunneling (SET) technology offers a potential for (sub)nanometer feature size scaling, room temperature operation, as well as ultra-low power consumption. However, similarly to other future technology candidates, it displays a switching behavior that differs from traditional MOS devices. This provides new possibilities and challenges for implementing computer arithmetic circuits.

In this paper, we investigated the implementation of basic arithmetic functions, such as addition and multiplication, in SET technology via the controlled transport of charge. First, we proposed a set of building blocks, e.g., move charge block ($MVke$) and periodic symmetric function block (PSF), which can be utilized for charge controlled computations. Second, using the new set of building blocks, we proposed a number of novel approaches for computing addition related

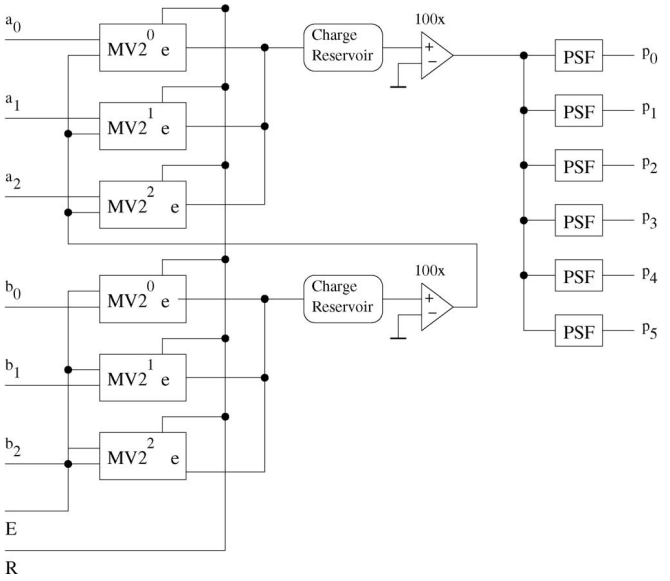


Fig. 17. 3-bit multiplier.

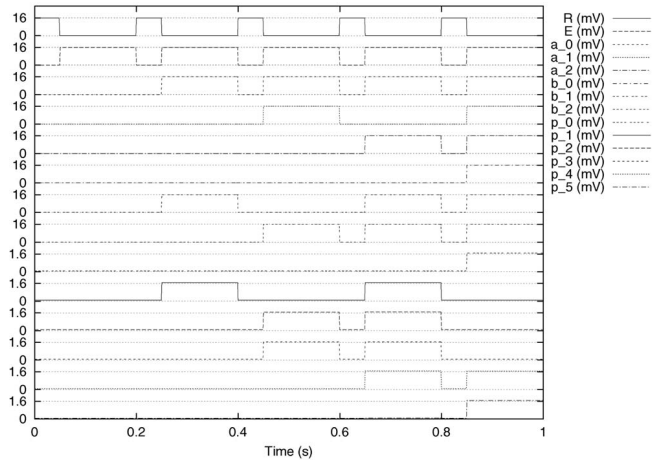


Fig. 18. 3-bit multiplier—simulation results.

arithmetic functions, e.g., addition, multiplication, via the controlled transport of charge. When the number of the electrons that can be accurately controlled within the system is unrestricted, we proved that the following holds true: The addition/subtraction of two n-bit operands can be computed with a depth-2 network composed out of $3n+1$ circuit elements; the multiplication of two n-bit operands can be computed with a depth-3 network with $4n-1$ circuit elements. For the more practical case when the number of the electrons that can be accurately controlled by an $MVke$ block is limited to 2^r-1, we proved that the addition/subtraction of two n-bit operands can be computed with a $\text{depth-}\left(\frac{n}{r}+3\right)$ network composed of $3n+1+\frac{n}{r}$ circuit elements. Additionally, under the same restriction, we suggested methods to reduce the addition network depth in the order of $\log\frac{n}{r}$ and to perform n-bit multiplication in an $O(\log\frac{n}{r})$ delay. Third, we introduced SET implementations for the electron counting blocks and evaluated their area and potential delay. The $MVke$ can be implemented with seven circuit elements and has a delay that varies between 5 and 10 ns, which means that the maximum delay is about the same with the typical delay of a CMOS-like gate, but $10\times$ larger than the one of a SEEL gate. The PSF block can be implemented with 13 circuit elements and has a delay that varies between 0.2 and 1.2 ns. Finally, we proposed SET-based implementations for a number of circuits operating under the electron counting paradigm as follows: 4-bit Digital to Analog Converter, 5-bit Analog to Digital Converter, 4-bit adder, and 3-bit multiplier.

The main advantage of the electron counting paradigm is the potential to encode an n-bit binary number as a single variable. First, this can result in a large reduction of area for memory cell arrays as well as for arithmetic circuits. Second, it can potentially result in reduced delay for arithmetic operations as electron counting logic can reduce/eliminate the carry chain that usually determines the critical path of such operations.

The main disadvantage of the electron counting paradigm is the need for additional signal amplification. Given that the charge present in a charge reservoir can potentially vary over a large range, the capacitance of the charge reservoir should be relatively large in order to reduce feedback to the attached electron counting building blocks. This also implies that the feed forward signal is relatively small and that it requires amplification. As this signal is non-Boolean, a simple buffer such as an inverter cannot be utilized. Instead, the presence of Opamp-like buffers is required. It may, however, be possible to delay signal amplification until a charge encoded result is converted into a binary number such that an inverter chain is sufficient for signal-level restoration.

Our investigation demonstrated the potential benefits the electron counting paradigm might have in terms of required area and delay for addition related operations. However, we do not yet have sufficient simulation data to evaluate it in relevant practical cases. The expected delay of the electron counting basic building blocks is larger than the one of SEEL gates, but we expect that the shallow networks produced by the electron counting paradigm can compensate for this. For example, when considering the n-bit addition, any fast structure based on carry lookahead or another similar technique [25] requires a delay in the order of $O(\log n)$, whereas the electron counting produces a network with a depth on the order of $O(\log\frac{n}{r})$ when the number of electrons that can be accurately controlled by an $MVke$ block is limited to 2^r-1. Whether or not this is enough to compensate for the larger delay of the electron counting blocks and/or for other practical issues that might limit the number of bits that can be accommodated within a charge reservoir is still an open issue and the subject of future research. However, given that the delay of electron counting blocks is about the same as the typical delay of a SEEL gate in the case of the PSF blocks and about one order of magnitude larger for the $MVke$ blocks, we expect adder and multiplier implementations based on the electron counting paradigm to be faster than SEEL-based designs. We also expect that the required area for addition related operations implemented in the electron counting paradigm will be lesser than the one required by SEEL implementation based on Boolean and/or threshold gates. When assuming that signal amplification can be achieved with an inverter chain, the power consumption might be

comparable to the SEEL approach, but this issue also requires more future investigations. Concluding, regarding the SET implementation of arithmetic operations, electron counting logic appears to be the design style that best exploits the SET potential. Another potential interesting application for this encoding scheme is the implementation of memory cell arrays as a large number of memory cells can utilize a single DAC and ADC.

REFERENCES

[1] Y. Taur, D. Buchanan, W. Chen, D. Frank, K. Ismail, S. Lo, G. Sai-Halasz, R. Viswanathan, H. Wann, S. Wind, and H. Wong, "CMOS Scaling into the Nanometer Regime," *Proc. IEEE,* vol. 85, no. 4, pp. 486-504, 1997.

[2] "Technology Roadmap for Nanoelectronics," http://www.cordis.lu/esprit/src/melna-rm.htm, 1999, published on the internet by the Microelectronics Advanced Research Initiative (MELARI NANO), a European Commission (EC) Information Soc. Technologies (IST) program on Future and Emerging Technologies.

[3] K. Likharev, "Single-Electron Devices and Their Applications," *Proc. IEEE,* vol. 87, no. 4, pp. 606-632, Apr. 1999.

[4] Y. Ono, Y. Takahashi, K. Yamazaki, M. Nagase, H. Namatsu, K. Kurihara, and K. Murase, "Fabrication Method for IC-Oriented Si Single-Electron Transistors," *IEEE Trans. Electron Devices,* vol. 49, no. 3, pp. 193-207, Mar. 2000.

[5] C. Wasshuber, "About Single-Electron Devices and Circuits," PhD dissertation, TU Vienna, 1998.

[6] C. Heij and J.M.P. Hadley, "A Single-Electron Inverter," *Applied Physics Letters,* vol. 78, no. 8, pp. 1140-1142, Feb. 2001.

[7] K. Ishibashi, D. Tsuya, M. Suzuki, and Y. Aoyagi, "Fabrication of a Single-Electron Inverter in Multiwall Carbon Nanotubes," *Applied Physics Letters,* vol. 82, no. 19, pp. 3307-3309, Feb. 2001.

[8] D. Averin and A. Odintsov, "Macroscopic Quantum Tunneling of the Electric Charge in Small Tunnel Junctions," *Physics Letters A,* vol. 140, no. 5, pp. 251-257, Sept. 1989.

[9] D. Averin and Y. Nazarov, "Virtual Electron Diffusion during Quantum Tunneling of the Electric Charge," *Physical Rev. Letters,* vol. 65, no. 19, pp. 2446-2449, Nov. 1990.

[10] S. Lotkhov, H. Zangerle, A. Zorin, and J. Niemeyer, "Storage Capabilities of a Four-Junction Single-Electron Trap with an On-Chip Resistor," *Applied Physics Letters,* vol. 75, no. 17, pp. 2665-2667, Oct. 1999.

[11] A. Zorin, S. Lotkhov, H. Zangerle, and J. Niemeyer, "Coulomb Blockade and Cotunneling in Single Electron Circuits with On-Chip Resistors: Towards the Implementation of the R Pump," *J. Applied Physics,* vol. 88, no. 5, pp. 2665-2670, Sept. 2000.

[12] S. Lotkhov, S. Bogoslovsky, A. Zorin, and J. Niemeyer, "Operation of a Three-Junction Single-Electron Pump with On-Chip Resistors," *Applied Physics Letters,* vol. 78, no. 7, pp. 946-948, Feb. 2001.

[13] K. Likharev, "Correlated Discrete Transfer of Single Electrons in Ultrasmall Tunnel Junctions," *IBM J. Research and Development,* vol. 32, no. 1, pp. 144-158, Jan. 1988.

[14] J. Tucker, "Complementary Digital Logic Based on the 'Coulomb Blockade'," *J. Applied Physics,* vol. 72, no. 9, pp. 4399-4413, Nov. 1992.

[15] R. Chen, A. Korotkov, and K. Likharev, "Single-Electron Transistor Logic," *Applied Physics Letters,* vol. 68, no. 14, pp. 1954-1956, Apr. 1996.

[16] N. Yoshikawa, Y. Jinguu, H. Ishibashi, and M. Sugahara, "Complementary Digital Logic Using Resistively Coupled Single-Electron Transistor," *Japanese J. Applied Physics,* vol. 35, no. 2B, pp. 1140-1145, Feb. 1996.

[17] M. Jeong, Y. Jeong, S. Hwang, and D. Kim, "Performance of Single-Electron Transistor Logic Composed of Multi-Gate Single-Electron Transistors," *Japanese J. Applied Physics,* vol. 36, no. 11, pp. 6706-6710, Nov. 1997.

[18] K. Likharev and V. Semenov, "Possible Logic Circuits Based on the Correlated Single-Electron Tunneling in Ultrasmall Junctions," *Extended Abstracts, Int'l Superconductive Conf.,* p. 182, 1987.

[19] Y.N. Nazarov and S.V. Vyshenskii, "SET Circuits for Digital Applications," *Single-Electron Tunneling and Mesoscopic Devices,* H. Koch and H. Lubbig, eds., vol. 31, pp. 61-66, Springer-Verlag, 1992.

[20] C. Lageweg, S. Cotofana, and S. Vassiliadis, "Static Buffered SET Based Logic Gates," *Proc. Second IEEE Conf. Nanotechnology (NANO),* pp. 491-494, Aug. 2002.

[21] C. Lageweg, S. Cotofana, and S. Vassiliadis, "Single Electron Encoded Latches and Flip-Flops," *IEEE Trans. Nanotechnology,* vol. 3, no. 2, pp. 237-248, June 2004.

[22] S. Cotofana, C. Lageweg, and S. Vassiliadis, "On Computing Addition Related Arithmetic Operations via Controlled Transport of Charge," *Proc. 16th IEEE Symp. Computer Arithmetic,* pp. 245-252, June 2003.

[23] C. Lageweg, S. Cotofana, and S. Vassiliadis, "Digital to Analog Conversion Performed in Single Electron Technology," *Proc. First IEEE Conf. Nanotechnology (NANO),* Oct. 2001.

[24] S. Vassiliadis, S. Cotofana, and K. Bertels, "2-1 Addition and Related Arithmetic Operations with Threshold Logic," *IEEE Trans. Computers,* vol. 45, no. 9, pp. 1062-1068, Sept. 1996.

[25] B. Parhami, *Computer Arithmetic—Algorithms and Hardware Design,* first ed. Oxford Univ. Press, 2000.

Sorin Cotofana received the MS degree in computer science from the Politehnica University of Bucharest, Romania, and the PhD degree in electrical engineering from Delft University of Technology (T.U. Delft), The Netherlands. He worked for a decade with the Research & Development Institute for Electronic Components (ICCE) in Bucharest. His work experience at ICCE was related to structured design of digital systems, design rule checking of IC's layout, logic and mixed-mode simulation of electronic circuits, testability analysis, and image processing. He is currently an associate professor on the Faculty of Electrical Engineering, Mathematics, and Computer Science of Delft University of Technology, The Netherlands. His research interests include computer arithmetic, parallel architectures, embedded systems, nanotechnology, reconfigurable computing, computational geometry, and computer-aided design. He is a senior member of the IEEE and a member of the IEEE Computer Society

Casper Lageweg received the MSc degree in electrical engineering from Delft University of Technology (T.U. Delft), The Netherlands. After completing his degree, he worked for Hewlett-Packard laboratories in Bristol, United Kingdom. Currently, he is working toward the PhD degree in the Computer Engineering Laboratory of the Delft University of Technology, The Netherlands. His research interests include nano electronics, single electron tunneling, logic design, computer arithmetic, computer architecture, integrated circuits, and physical design. He is a student member of the IEEE.

Stamatis Vassiliadis is a professor on the Faculty of Electrical Engineering, Mathematics, and Computer Science of Delft University of Technology, The Netherlands. He has also served on the faculties of Cornell University, Ithaca, New York, and the State University of New York (SUNY), Binghamton. He worked for a decade with IBM in the Advanced Workstations and Systems laboratory in Austin, Texas, the Mid-Hudson Valley laboratory in Poughkeepsie, New York, and the Glendale laboratory in Endicott, New York. At IBM, he was involved in a number of projects regarding computer design, organizations, and architectures and the leadership to advanced research projects. For his work, he received numerous awards, including 23 levels of Publication Achievement Awards, 15 levels of Invention Achievement Awards, and an Outstanding Innovation Award for Engineering/Scientific Hardware Design in 1989. He is a fellow of the IEEE and a member of the IEEE Computer Society.