Computing Division Using Single-Electron Tunneling Technology

Cor Meenderinck, *Student Member, IEEE*, and Sorin Cotofana, *Senior Member, IEEE*

***Abstract*—Emerging nanotechnologies, like single-electron tunneling (SET) technology, possesses properties that are fundamentally different from what CMOS offers to engineers. This opens up avenues for novel computational paradigms, which can perform arithmetic operations efficiently by utilizing these new available properties. In this line of reasoning, in this paper we investigate the implementation of division in SET technology using a novel computation paradigm called electron counting. First, we present two schemes that are based on sequential approximation of the quotient. The first scheme is basic and simple to build, but suffers from overshoot and has a rather large delay. The second scheme, which is a modification of the first one, has a delay logarithmic in the quotient magnitude and the simulation results we present indicate that this scheme works correctly. Finally, we propose a division scheme based on the computation of periodic symmetric functions. Although this scheme requires a varactor for which no nanoscale implementation yet exists and which cannot be directly simulated, it demonstrates the possibilities that nanotechnology, and specifically SET technology, potentially offers as it has a time complexity of $O(1)$.**

***Index Terms*—Computer arithmetic, division, single-electron tunneling.**

I. Introduction

FOR DECADES we have seen an ongoing increase in integrated circuit performance mainly due to advances in fabrication technology and improvements in computational paradigms. However, fabrication technology is stagnating and it is generally expected that current technology, i.e., CMOS, cannot be pushed beyond a certain limit. This limit is expected to arise in mainly two areas: power consumption and scalability. The International Technology Roadmap for Semiconductors (ITRS) [1] states that ”we have reached the point where the horizon of the Roadmap challenges the most optimistic projections for continues scaling of CMOS.” Consequently, the roadmap included post-CMOS devices. Moreover, the amount of research in this emerging technologies field has exploded.

A promising candidate to succeed CMOS is single-electron tunneling (SET) technology [2], as it does not suffer from the limitations CMOS faces (power consumption and scalability). SET technology allows controlling single or few electrons and therefore has potential to perform computation with ultra low power consumption. Downscaling feature sizes increases the quantum mechanical behavior, especially when reaching the nanometer region. For CMOS this causes problems, whereas for SET, which is based on the quantum nature of tunnel transport, this improves device behavior. Consequently, SET technology is scalable to the nanometer region and beyond.

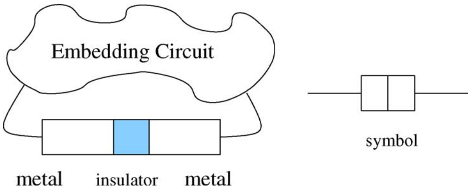


Fig. 1. Schematic representation of the tunnel junction.

Several proposals have been made to implement computational operations using SET technology and these implementations are mainly categorized in two types (see, for example, [2] and [3]). The first type of implementation represents logic values by voltage [4]–[6] and SET devices are used to switch currents on and off, resulting in MOSFET-like behavior. The main advantage of this approach is that current designs can easily be translated into SET technology, but there are disadvantages as well. A major drawback to this approach is the large power dissipation [7], [8]. The second type of implementation represents bits by single or few electrons and consequently consumes less power. Single-electron encoded logic (SEEL) [9] is an example of the latter.

Using the second type of implementation, arithmetic units can be designed in a conventional logic design style, e.g., using Boolean and/or threshold gates (see, for example, [10]). The electron counting (EC) paradigm [11], on the other hand, uses a novel design style and appears promising as an efficient methodology for implementing SET-based arithmetic operations. In previous research, addition-related arithmetic operations, i.e., addition [12], [13] and multiplication [12], [14] have been implemented utilizing the EC paradigm.

In this paper we investigate the implementation of EC-based division. First, Section II briefly describes the single-electron tunneling phenomenon and introduces the EC paradigm. Second, using previously designed EC building blocks, in Section III we present a basic division scheme, which has a delay linear in the magnitude of the quotient. Subsequently, to improve circuit delay, in Section IV an improved division scheme is proposed, the implementation of which is explained in detail, and simulation results are presented. This scheme has a delay that is logarithmic in the magnitude of the quotient. Third, in Section V we propose a novel division scheme, based on the computation of periodic symmetric functions, and which has a time complexity of $O(1)$. Section VI concludes the paper.

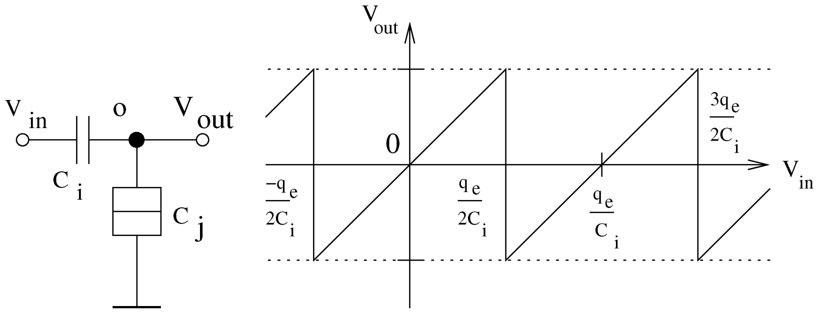
Manuscript received July 13, 2006; revised February 20, 2007. The review of this paper was arranged by Associate Editor K. Likharev.

The authors are with the Computer Engineering Laboratory, Delft University of Technology, Delft, the Netherlands (E-mail: Cor@ce.et.tudelft.nl; Sorin@ce.et.tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNANO.2007.901378

1536-125X/$25.00 © IEEE



(a) (b)

Fig. 2. SET electron trap. (a) Circuit. (b) Transfer function.

II. Background

SET circuits are based on tunnel junctions which consist of an ultrathin insulating layer in a conducting material (see Fig. 1). In classical physics no charge transport is possible through an insulator. However, when the insulating layer is thin enough the transport or *tunneling* of charge can be controlled in a discrete and accurate manner, i.e., one electron at a time. Tunneling through a junction becomes possible when the junction's current voltage V_j exceeds the junction's critical voltage [15] $V_c = q_e/2(C_e + C_j)$, where $q_e = 1.602 \cdot 10^{-19} C$, C_j is the capacitance of the junction, and C_e is the capacitive value of the remainder of the circuit as seen from the junction. In other words, tunneling can occur if and only if $|V_j| \geq V_c$.

Electron tunneling is stochastic in nature and as such the delay cannot be analyzed in the traditional sense. Instead, for each transported electron the switching delay can be described as

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

where R_t is the junction's resistance and P_{error} is the probability that the desired charge transport has not occurred after t_d seconds. In this paper we assume $R_t = 10^5 \Omega$ and $P_{\text{error}} = 10^{-8}$. Each transported electron reduces the system energy by $\Delta E = q_e(|V_j| - V_c)$ from which the consumed energy can be calculated.

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 $P_{\text{therm}} = e^{\Delta E/k_B T}$, where k_B is Boltzmann's constant ($k_B = 1.38 \cdot 10^{23}$ J/K), T is the operating temperature, and ΔE is the change in the systems energy 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 [16], [17]. The cotunneling error probability can be reduced sufficiently through the addition of strip resistors [18]–[20]. 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.

One issue that seems to hurt any nanoscale technology, whether it is CMOS or SET, is variability. Several sources of variability are known and at least some of them, e.g., process parameter variations, will affect SET circuitry. In [21] an analysis of several SET gates was presented indicating that their robustness to variability can be low. Although this issue is out of the scope of this paper, we do acknowledge that design for variability and even design for fault tolerance are essential for effective utilization of the SET technology.

Note that the implementations discussed in here are technology independent. SET tunnel junctions can for example be implemented by classical semiconductor lithography and by carbon nanotubes [22]. Therefore, circuit area is evaluated in terms of the total number of circuit elements (capacitors and junctions).

A well-known SET structure is the electron trap depicted in Fig. 2(a). The SET electron trap functions as follows. If the input voltage rises, the output voltage follows due to capacitance division. At some point, though, the voltage across the tunnel junction exceeds the critical voltage and an electron tunnels to the output node. Consequently, the output voltage drops. As the input voltage continues to rise, the output voltage rises again until it reaches the critical voltage, etc. Thus, the electron trap has a periodic transfer function as depicted in Fig. 2(b).

The relation between the input voltage V_{in} and the output voltage V_{out} of the electron trap can be derived as

$$V_{\text{out}} = \frac{C_i}{C_{\Sigma o}} V_{\text{in}} - \frac{q_o}{C_{\Sigma o}} \tag{2}$$

where $C_{\Sigma o}$ is the sum of all capacitances connected to node o and q_o is the net charge in node o. Also, the input voltage for which the output voltage reaches its maximum can be expressed as

$$V_{\text{in,peak}} = \frac{q_e}{2C_i} + \frac{kq_e}{C_i} \qquad \text{for } k = 0, 1, 2, \ldots. \tag{3}$$

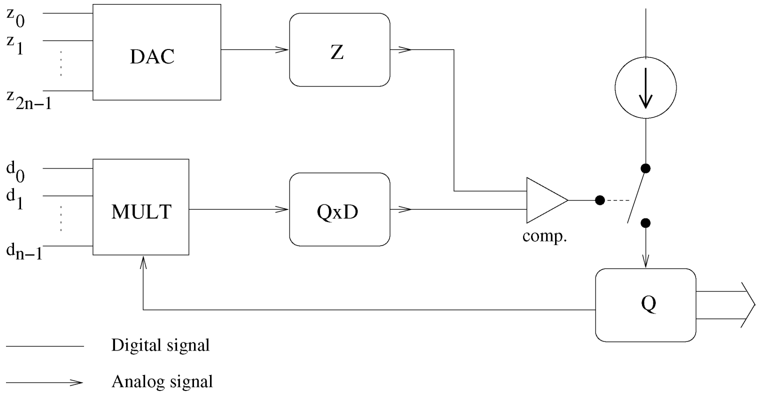


Fig. 3. Basic quotient approximation division scheme. Dividend z is stored in analog format in reservoir Z. The quotient (Q) starts at zero and is increased by adding charge to reservoir Q. Once $Q \times D \geq Z$ (D is the divisor) the comparator switches of the current source. Reservoir Q contains the resulting quotient.

This equation shows that the period of the electron trap transfer function is only dependent on the magnitude of capacitance C_i. The periodic nature of the electron trap transfer function provides the fundamentals for effective implementation of periodic symmetric functions (PSFs) in SET technology, which is further explained in Section IV-B2. PSFs are an essential element for implementing arithmetic operations in the EC paradigm.

There are many ways to do computation using SET technology, but most do not seem to fully utilize the potential offered by SET. The EC paradigm is a novel way of computation that does exploit the potential of SET technology to a greater extend. In the EC paradigm, the ability to control the transport of individual electrons is utilized to encode an integer value X directly as a charge Xq_e. Using a digital to analog converter (DAC) [12], a binary value can be converted into an amount of charge and stored on a relatively large capacitor (a charge reservoir). Once binary values have been encoded as a number of electrons, arithmetic operations can directly be performed in electron charges, which reveals a broad range of novel computational schemes. The result of the arithmetic operations can be converted back to a binary value using an analog to digital converter (ADC) [12]. Based on this approach, in previous research [12], [14] we implemented addition and multiplication schemes, using respectively a depth-2 and a depth-3 network. In this paper we investigate the implementation of a nonlinear arithmetic operation, i.e., division, in the EC paradigm. In the next section we first present a simple EC division scheme, which is based on quotient approximation.

III. Basic Quotient Approximation Division

Binary division is defined as $z = (q \times d) + s$ with $s < d$, where z is the dividend, d is the divisor, q is the quotient, and s is the remainder. Assuming an operand size of n bits for d, generally speaking the operand size for z is $2n$, resulting in an n-bit quotient (q) and an n-bit remainder (s). Such a divider is referred to as a $2n$-bit by n-bit divider.

First, in the next section we present a block level description of the basic quotient approximation division scheme and explain its functionality. Second, in Section III-B we present detailed descriptions of the building blocks utilized by the division scheme.

A. Strategy

A basic way to make an EC-based division scheme is presented in Fig. 3 and it works as follows. The dividend z, which we assume to be binary encoded, is converted by a DAC scheme into an analog value, i.e., a number of electrons stored in charge reservoir Z. The bits of the divisor d are connected to an EC multiplier [12], which computes the product of d and the quotient q, and stores this value in charge reservoir $Q \times D$. The quotient q, stored in charge reservoir Q, is reset to zero before the start of each computation. A comparator keeps track of the values of Z and $Q \times D$ and allows a current source to subtract electrons from reservoir Q as long as $Q \times D$ is smaller than Z. In our EC-based circuit implementations we represent values by positive charge, meaning that removing electrons from a charge reservoir increases the represented value. When the value in reservoir $Q \times D$ is equal to or greater than the value in Z, the comparator opens the switch and the removal of electrons from reservoir Q stops. The final result of the division is in charge reservoir Q and, if necessary, can be converted back to the digital domain by utilizing an *ADC* scheme. We note here that for noninteger quotients this scheme always rounds off upward.

B. Implementation

The division scheme utilizes a DAC building block and an EC multiplier, both consisting of a parallel structure of MVke blocks (MoVe k electrons block), resulting in a depth-1 network. For a detailed description of the implementations we refer the reader to [12].

The comparator was implemented using an SET threshold logic gate (TLG). A threshold gate is a device that computes a function given by:

$$F(X) = sgn\{\mathcal{F}(X)\} = \begin{cases} 0 & \text{if } \mathcal{F}(X) < 0 \\ 1 & \text{if } \mathcal{F}(X) \geq 0 \end{cases} \tag{4}$$

$$\mathcal{F}(X) = \sum_{i=1}^{n} \omega_i x_i - \phi \tag{5}$$

where x_i are the inputs, w_i are the corresponding weights, and ϕ is an internal threshold value. For this scheme we used $\mathcal{F}(X) = Z - Q \times D$, thus it can be implemented by a two input threshold gate, with $X = \{Z, Q \times D\}$, $W = \{1, -1\}$ and $\phi = 0$. For more details on the implementation of the SET threshold gate we refer the reader to [23].

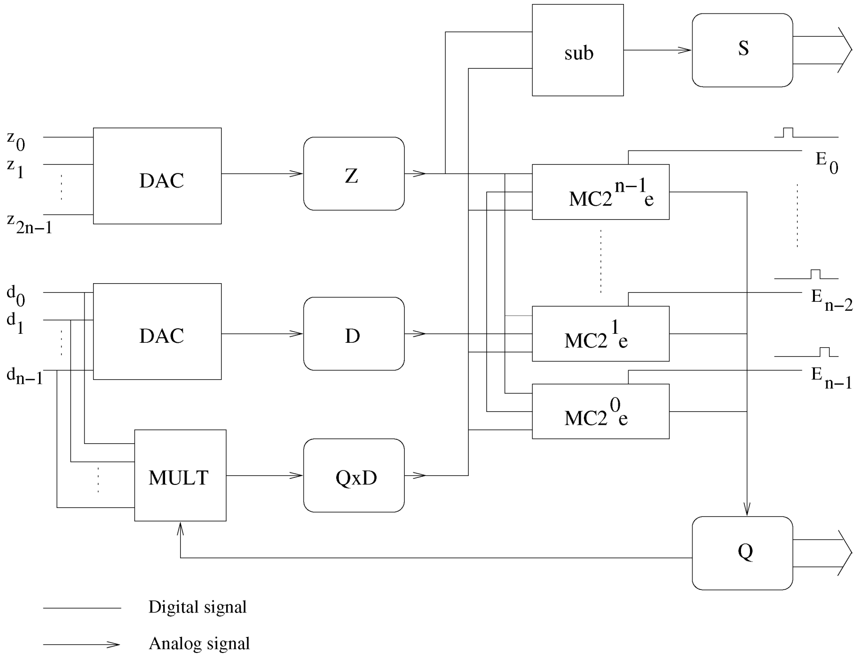


Fig. 4. Improved quotient approximation division scheme. The quotient Q starts at zero. Each $MC2^ie$ block, subsequently enabled from $i = n - 1$ down to 0, removes 2^i electrons from Q if $Z - Q \times D > 2^iD$. Reservoir S contains the remainder.

The current source and switch were implemented by a single-electron transistor (see [5] for an early analysis), which operates similar to a MOS transistor. However, the current through a single-electron transistor consists of discrete electrons that tunnel strictly one after another, allowing us to control the charge in an accurate manner.

The basic division scheme, as presented so far, does not compute the remainder. Although this could in principle be implemented, we actually did not do so because of the resulting increase in delay and area costs.

C. Simulation

We implemented the division scheme in Fig. 3 in the simulation environment SIMON [24]. In the simulations that we performed the circuit suffered from heavily overshoot, i.e., the single-electron transistor was closed too late resulting in a too large charge in reservoir Q. That problem can be solved by clocking the transfer of electrons to the charge reservoir Q, thus by adding a delay in the feedback loop. This could be implemented by replacing the single-electron transistor with an electron pump [25]. We did not simulate this possibility as it would worsen the delay of the scheme, which already is rather large. Either using an SET transistor or an electron pump, electrons are removed from reservoir Q one after another. Therefore, the delay of this division scheme is linear in the magnitude of the quotient.

Concluding, this first division scheme is simple but has a rather large delay. In its basic form it does not compute the remainder, which could be resolved but requires quite some hardware and annuls the only positive asset of this scheme, i.e., its simplicity.

IV. Improved Quotient Approximation Division

The delay of the basic EC division scheme, as presented in the previous section, can be reduced by transferring electrons in groups. This observation led to the design of the improved EC division scheme, presented in this section. Again, first we present the block level description of the circuit and its strategy after which we discuss the used building blocks in detail.

A. Strategy

The analog value of Q can be described as a sum of powers of two, i.e., $Q = \sum_{i=0}^{n-1} q_i 2^i$, where q_i is a Boolean coefficient. By determining the Boolean coefficients q_i and removing the corresponding number of electrons we can compute Q.

The EC division scheme as depicted in Fig. 4 uses this strategy and works as follows. Assuming an n-bit quotient, the addition of charge to reservoir Q is performed by n parallel MCke (move conditional k electrons) building blocks. These blocks are consecutively enabled, starting with the most significant one. MCke block i ($i = n - 1, n - 2, \ldots, 0$) removes $k = 2^i$ electrons from reservoir Q if the condition $\Psi_i = (Z - Q \times D > 2^iD)$ is true. That is, it computes, based on the current estimation of Q, whether the removal of k electrons from Q would result in an estimation of Z (that is $Q \times D$) that is smaller or larger than the actual Z value. If the estimation is calculated as being lower, the MCke block removes k electrons from Q, otherwise it does not. Consequently, after the last MCke block has been enabled reservoir Q contains the quotient. We note here that for noninteger quotients this scheme rounds off downward.

The remainder, defined as $s = z - q \times d$, can easily be computed in this scheme using a subtraction block. The subtraction block uses as inputs the analog values from reservoirs Z and $Q \times D$ and stores the remainder in reservoir S.

As an example Table I describes the operation of the division scheme for $Z = 14$ and $D = 4$. For this case three MCke blocks are required that have parameters $k = 4$, $k = 2$ and $k = 1$. The operation starts with $Q = 0$ and the first estimation of Z, i.e., $Q \times D$, is 0. The difference between Z and its estimation is

TABLE I
THE ALGORITHM FOR $Z = 14$ AND $D = 4$.

Q	QxD	$Z - Q$xD	i	2^iD	$Z - Q$x$D - 2^iD$	Ψ
0	0	14	2	16	-2	FALSE
0	0	14	1	8	6	TRUE
2	8	6	0	4	2	TRUE
3						

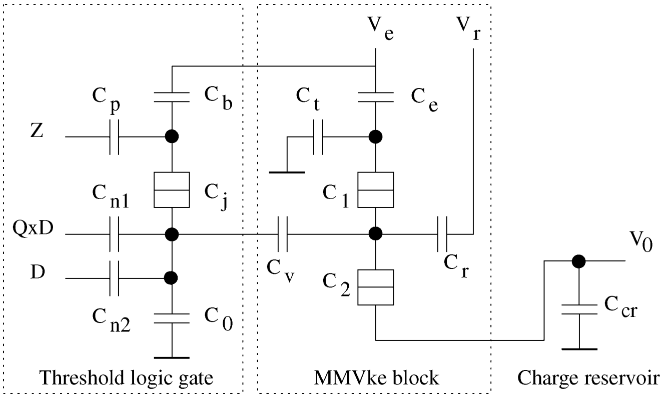


Fig. 5. Modified MCke building block. The threshold logic gate evaluates $Z - Q \times D > kD$, and if true the MMVke block removes k electron, thus adds charge to the charge reservoir.

14, which is smaller than 2^iD. Thus, condition Ψ_2 is FALSE and the first MCke block removes no electrons from reservoir Q. Subsequently, the second MCke block evaluates condition Ψ_1, resulting in TRUE, so the second MCke block removes 2 electrons from Q. Finally, the last MCke block evaluates condition Ψ_0 using the new value for Q resulting in TRUE, therefore removing one electron from Q. The final result is $Q = 3$, which is the correct result of the division.

Given that the evaluation of the conditions Ψ_i has to be done serially, the delay of the circuit is determined by the number of MCke blocks. Thus, it is logarithmic in the maximum magnitude of the quotient.

B. Implementation

The improved quotient approximation division scheme requires some additional functional blocks when compared to the basic version. In this section we present the implementations of these blocks, i.e., the MCke block, the *ADC* block, the subtraction block (which utilizes the *ADC* block), and the control logic.

1) The MCke Block: An MCke building block implementation was proposed in [14]. The MCke building block consists of two parts, an SET threshold logic gate and an MMVke block. The first part is responsible for evaluating condition Ψ_i on the input signals, while the second part is responsible for removing electrons from the charge reservoir connected to the output. Condition Ψ_i can be expressed as $\mathcal{F}(X) = Z - Q \times D - 2^iD$, thus it can be implemented by a three input threshold gate, with $X = \{Z, Q \times D, D\}$, $W = \{1, -1, -2^i\}$ and $\phi = 0$. Since the original MCke implementation had only one input and could only evaluate simple conditions, we replaced the threshold gate in the first stage with a three input version, resulting in the implementation in Fig. 5.

The MMVke block is a modified version of the MVke block. It also removes Vk electrons when enabled and has a similar structure but internally works a little differently, which makes it suitable to be used in combination with the threshold gate. The implementation of the MMVke block is discussed in detail in [14].

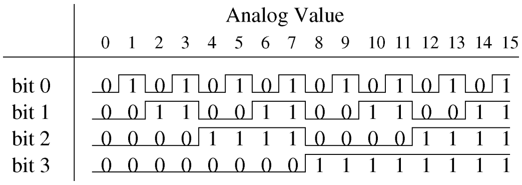

Analog Value
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
bit 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
bit 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
bit 2 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
bit 3 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

Fig. 6. AD conversion using PSFs. Each bit i can be expressed as a PSF with period 2^{i+1}.

2) The ADC Block: To build the subtraction block an ADC block is required, which was first introduced in [11]. The functionality of the EC analog to digital conversion block is based on PSFs, which we first explain. An n-variables function is symmetric if and only if for any permutation σ of $\langle 1, 2, \ldots, n \rangle$, $F_s(x_1, x_2, \ldots, x_n) = F_s(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})$. In other words, a symmetric function is independent on the order of its input operands.

As Boolean functions have operands that are either "0" or "1" Boolean symmetric functions depend on the number of "ones" in the input. Thus, a Boolean symmetric function entirely depends on the sum of its input values: $F_s(x_1, x_2, \ldots, x_n) = F_s(\sum_{i=1}^{n} x_i)$. This allows for a more compact representation of the function as it can be described by a vector $v = v_0 v_1 \ldots v_n$ where v_i is the output of F_s when the sum of the inputs is i. This representation is linear in the number of inputs while the traditional truth table has a size of 2^n entries.

A generalized symmetric function $F_g(X)$ is a function that depends on the weighted sum of its inputs $X = (\sum_{i=1}^{n} x_i w_i)$, where w_i is the weight of input x_i.

A PSF is a symmetric function for which there exists a period T such that $F_s(X) = F_s(X + T)$. A PSF is completely defined by the constants a, b, and T, where a is the first positive transition and b is the first negative transition.

In the analog to digital conversion process, each output bit can be described as a PSF of the analog input value. Fig. 6 depicts 4-bit AD conversion both as logic values and as a waveform. The waveform shows that each output bit b_i is a PSF of the input with a period of 2^{i+1}. Thus, an n-bit ADC block can be implemented using n parallel functional blocks that compute PSFs.

Such a functional block, referred to as a PSF block, can be implemented using the electron trap depicted in Fig. 2, as a basis. The electron trap has a periodic transfer function and the period can be adjusted through the value of capacitance C_i. To transform the sawtooth shaped transfer function of the electron trap into the rectangular shaped PSF, an SET inverter [9] can be utilized, which then acts as a literal gate. The full implementation of the PSF block is depicted in Fig. 7.

3) The Subtraction Block: In order to compute the remainder a subtraction block is required. One possible implementation is depicted in Fig. 8. In this implementation, the subtraction is performed by adding the value of $Q \times D$ to the reservoir S while subtracting the value of Z from it, thus computing a negative remainder. Adding the value of $Q \times D$ is done by means of an MVke block (with $k = 1$), where the V input is directly connected to reservoir $Q \times D$. Theoretically, removing the value

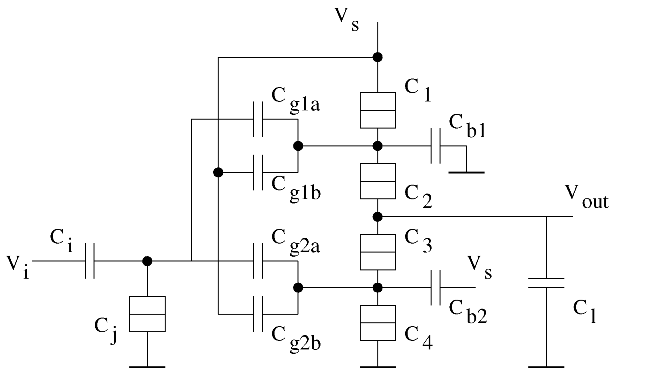


Fig. 7. Implementation of the PSF block. C_i and C_j form an electron trap which has a sawtooth shaped period transfer function. The rest of the circuit forms a buffer which acts as a literal gate.

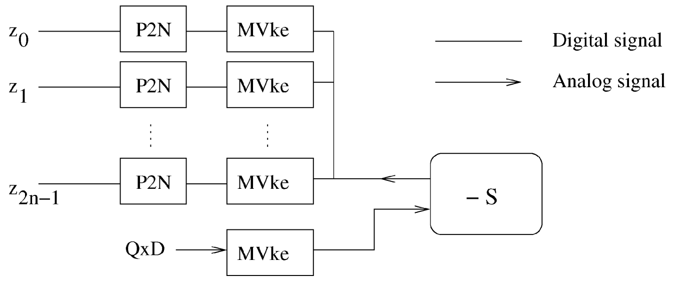


Fig. 8. Overview of the subtraction block. $Q \times D$ is added to the reservoir by a normal MVke block. The binary signals z_i are first converted from positive to negative voltage. MVke blocks, powered with negative supply voltage, add the corresponding number of electrons to the reservoir, and thus subtract the value of Z.

of Z can be done in a similar way but as of yet no building block is available that can do so. Instead we apply a strategy which utilizes the binary inputs of Z and has a structure similar to the DAC block. In normal operation the MVke block removes electrons from a reservoir, thus increasing the value stored in it. In this case we apply negative voltages to all inputs (V, enable, and reset), causing it to add electrons to the reservoir and thus subtracting value.

For the inputs of the MVke blocks we use the binary signals $\{z_0, z_1, \ldots, z_{2n-1}\}$ and not the analog value from reservoir Z as suggested by Fig. 4. However, those binary signals are represented by positive voltages. This problem was solved by using positive to negative (P2N) blocks which transform a binary positive voltage into a binary negative voltage.

The P2N building block was implemented using an SET inverter connected to a negative supply voltage as depicted in Fig. 9 and works as follows. If the input voltage is zero, the circuit is charge neutral and the output voltage is zero too. If the input voltage becomes high, the voltage across junction 4 (C_4) exceeds its critical voltage and one electron tunnels from the supply voltage to node z. This in turn results in a voltage across junction 3 (C_3) that exceeds its critical voltage. Thus, one electron tunnels from node z to node y, resulting in a negative charge on capacitor C_o and thus a negative output voltage. If the input voltage changes to zero again, a similar sequence of tunnel events happens in respectively junction 1 and junction 2, which results in a zero output voltage.

4) The Control Logic: The linear division scheme requires a set of enable signals, which together have the shape of a pulse train. We propose to generate these signals by local control logic, more specifically by using a delay line structure, build out of serially cascaded SET inverters [9]. This even allows us to assign to each MCke block a different delay time, depending on the number of electrons it has to transfer.

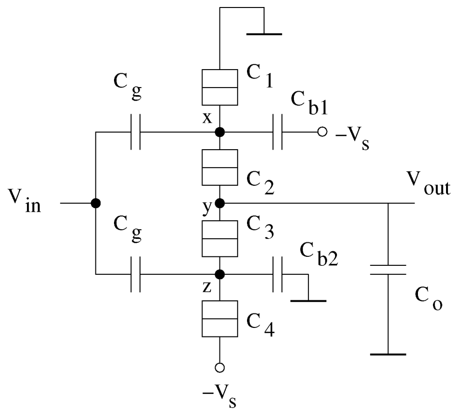


Fig. 9. The P2N building block. Binary inputs are converted from positive to negative voltage using a buffer powered with negative supply voltage.

C. Simulation

To verify the proposed linear EC-based division scheme we simulated a 6-bit by 3-bit instance. Fig. 10 presents the simulation results for 14/3 and 14/4. The first result corresponds to the numbers in Table I. The top two rows show the reset and (block level) enable signal, respectively. The third row shows the value of the remainder S. The next three rows are the (internal) enable signals for the MCke blocks and the last row represents the quotient Q. As the SIMON simulator does not provide any delay information, the time axis has no dimension. Below we present the worst case delay of the scheme based on (1). Furthermore, the control logic, i.e., the delay line, could not be simulated, due to the timing limitations of SIMON. When simulated in SIMON, a gate produces its output instantaneous when the inputs change. Thus, a line of serially cascaded inverters would all switch at the same time, instead of one after another. In order to simulate the division scheme we used three voltage sources to generate the enable signals for the MCke blocks.

We also simulated the circuit for other input values and all simulations indicate that this scheme functions correctly. The worst case delay was calculated as 50.5 ns and the worst case energy consumption was calculated as 4.1 eV. The area cost of the scheme is 363 elements.

The delay of this division scheme is determined by the number of steps in the approximation which is logarithmic in the maximum magnitude of the quotient. Reminding that the basic quotient approximation division scheme has a linear time complexity with respect to the magnitude of the quotient, the improved scheme is better in terms of time efficiency. However, it still does not seem to be the best SET technology can offer. In the next section we propose the implementation of an $O(1)$ division scheme.

V. Adjustable PSF Division

In this section we propose an implementation of an EC-based division scheme that has a time complexity of $O(1)$. The scheme is theoretical in the sense that it requires an element, i.e., a

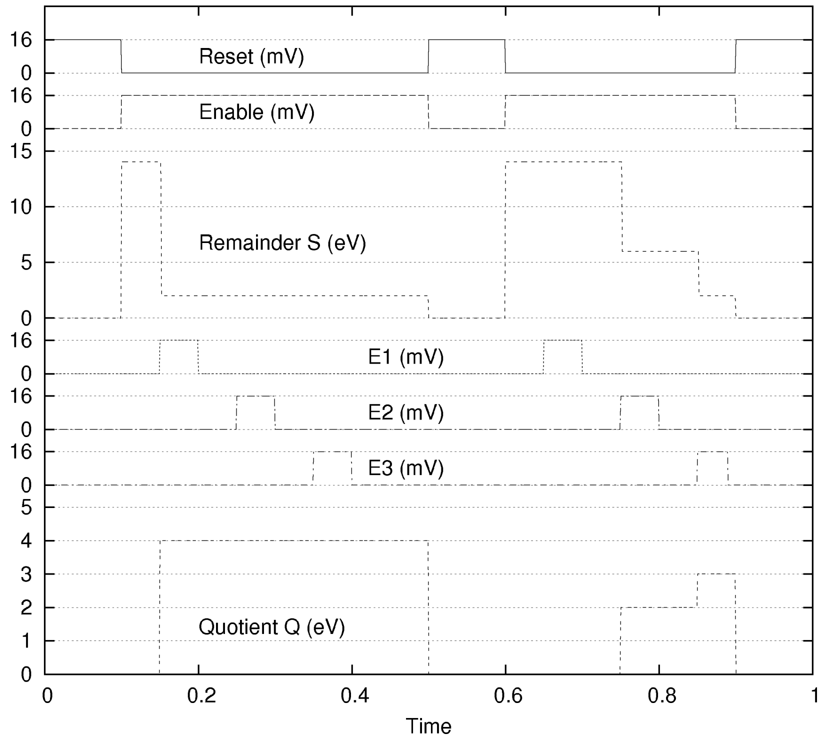


Fig. 10. Simulation results of the improved quotient approximation division scheme. At time 0.1 the calculation of 14/3 starts while at time 0.6 the calculation of 14/4 starts.

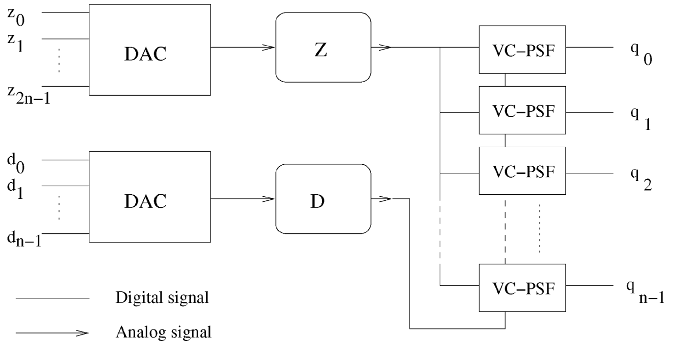
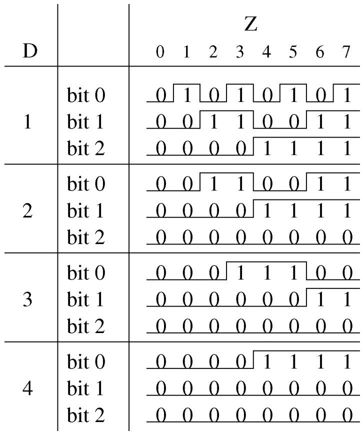


Fig. 11. Adjustable PSF division scheme. First, dividend Z and divisor D are converted from digital to analog signals. Next, the VC-PSFs convert Z back to the digital domain using a new scale which is inverse proportional to the value of D, such that the result is Q.



D		Z 0	1	2	3	4	5	6	7
1	bit 0	0	1	0	1	0	1	0	1
	bit 1	0	0	1	1	0	0	1	1
	bit 2	0	0	0	0	1	1	1	1
2	bit 0	0	0	1	1	0	0	1	1
	bit 1	0	0	0	0	1	1	1	1
	bit 2	0	0	0	0	0	0	0	0
3	bit 0	0	0	0	1	1	1	0	0
	bit 1	0	0	0	0	0	0	1	1
	bit 2	0	0	0	0	0	0	0	0
4	bit 0	0	0	0	0	1	1	1	1
	bit 1	0	0	0	0	0	0	0	0
	bit 2	0	0	0	0	0	0	0	0

Fig. 12. Division using an adjustable PSF. Bit i of Q is described as a PSF of Z with a period of $D2^{i+1}$.

voltage controlled capacitor or varactor, for which to the best of our knowledge no nanoscale implementation has so far been proposed. However, we believe this scheme has great potential and certainly shows the opportunities provided by SET technology.

The scheme we propose in this section is based on PSFs, which were explained in Section IV-B2. The general idea behind this division scheme is to dynamically adjust the period of the PSF blocks. As Fig. 11 depicts, the dividend is stored as an analog value in reservoir Z. Subsequently, that value is converted back to the digital domain using voltage-controlled PSF (VC-PSF) blocks where the period is dynamically adjusted through the value of D. Fig. 12 shows a simple example of how the period of the PSFs is changed according to the divisor value and how the analog to digital conversion results in the value of the quotient Q. For $D = 1$ the period of the PSF blocks is nominal and the AD conversion results in the digital representation of the value of Z. For $D = 2$ all periods of the PSF blocks are doubled and as a result the AD conversion outputs $Z/2$, e.g., $Q = 11_b$ for $Z \in \{6, 7\}$. For $D = 3$, the period is tripled, etc.

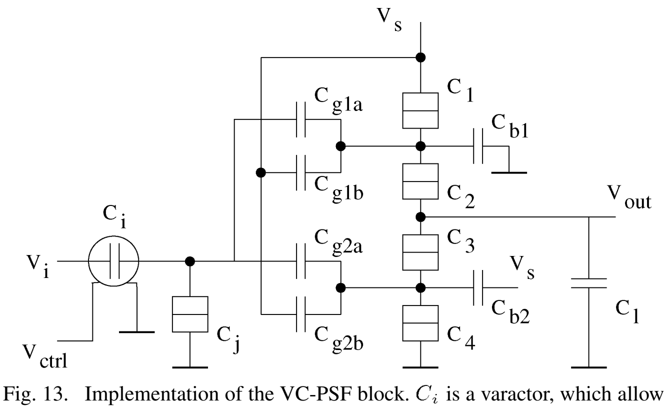


Fig. 13. Implementation of the VC-PSF block. C_i is a varactor, which allows to control the period of the VC-PSF block through voltage V_{ctrl}.

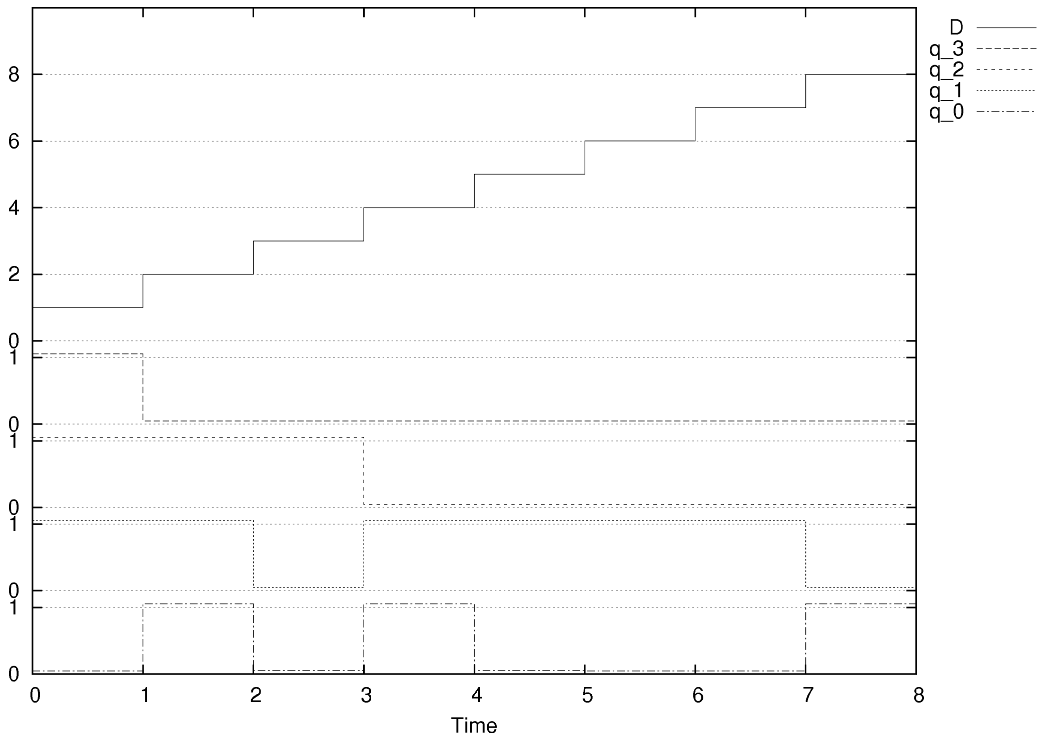


Fig. 14. Simulation results of the adjustable PSF division scheme for $Z = 14$.

To implement the adjustable PSF, we propose the implementation of a VC-PSF block as depicted in Fig. 13. The only difference with the normal PSF block is the capacitor C_i, which has been replaced with a voltage controlled capacitor (often referred to as a varactor). Although such a device has not been implemented at nanoscale, there exist techniques that potentially can do so. The most common way to implement a varactor is to use a diode. However, such a device is only suitable for operation on ac signals. Recently, ferroelectric thin-film devices [26] have been proposed. Using this new technology, which can easily be integrated in semiconductor technology, an implementation of a varactor was proposed by Subramanyam *et al.* [27]. Another recent development is using organic semiconductors to implement an organic voltage-controlled capacitor [28]. Thus, even though at time of writing no nanoscale implementation of a varactor is available, we expect that such a device can be implemented in the future.

The simulation environment SIMON does not support a varactor as a circuit element. Therefore, we could not directly simulate the division scheme. However, to demonstrate the potential of the scheme we simulated the varactor with manual adjustment of the value of capacitor C_i. The simulation results are depicted in Fig. 14 for $Z = 14$. The plot shows that indeed the output of the analog to digital converter equals the value of the quotient Q. We calculated, assuming a 6-bit by 3-bit divider, the worst case delay as 18 ns and the worst case energy consumption as 2.3 eV. For these calculations, the varactor was modelled as two separate capacitors. The area cost of the divider is 143 elements. On all three parameters this scheme scores better then the improved quotient approximation scheme. For larger operand sizes the difference will be even greater as the first has time complexity $O(1)$ while the latter has time complexity $O(n)$.

VI. Conclusion

In this paper we presented three EC-based division schemes in SET technology. The first two schemes are based on sequential approximation of the quotient. The first scheme is basic and simple to build, but suffers from overshoot and has a rather large delay. The second scheme, which is a modification of the first scheme, has linear time complexity and all simulations indicated that it works correctly. For a 6-bit by 3-bit division, we calculated the worst case delay of this scheme as 50.5 ns and the worst case energy consumption as 4.1 eV. Finally, we proposed a division scheme based on the computation of PSFs. For this scheme, which has an $O(1)$ time complexity, we computed, assuming the same operand sizes, a delay of 18 ns and an energy consumption of 2.3 eV. This performance improvement increases for large operand sizes. The proposed division schemes reveal the possibilities that nanotechnology, and specifically SET technology, potentially offers for computing arithmetic functions. The work presented in this paper indicates that if engineers are willing to leave behind the traditional way of computation and are willing to embrace new paradigms, nonlinear operations like division can be implemented using relative simple circuits which have excellent time characteristics.

References

[1] International Technology Roadmap for Semiconductors, 2005 Edition, Executive Summary 2005 [Online]. Available: http://www.itrs.net

[2] R. Waser, Ed., *Nanoelectronics and Information Technology—Advanced Electronic Materials and Novel Devices*, 1st ed. Berlin, Germany: Wiley-VCH, 2003.

[3] K. Likharev, “Single-electron devices and their applications,” *Proc. IEEE*, vol. 87, no. 4, pp. 606–632, Apr. 1999.

[4] J. Tucker, “Complementary digital logic based on the ‘coulomb blockade’,” *J. Appl. Phys.*, vol. 72, no. 9, pp. 4399–4413, Nov. 1992.

[5] K. Likharev, “Single-electron transistors: Electronic analogs of the dc squids,” *IEEE Trans. Magn.*, vol. MG-23, no. 2, pp. 1142–1145, Mar. 1987.

[6] R. Chen, A. Korotkov, and K. Likharev, “Single-electron transistor logic,” *Appl. Phys. Lett.*, vol. 68, no. 14, pp. 1954–1956, Apr. 1996.

[7] A. Korotkov, R. Chen, and K. Likharev, “Possible performace of capacitively coupled single electron transistors in digital circuits,” *J. Appl. Phys.*, vol. 78, no. 4, pp. 2520–2530, Aug. 1995.

[8] D. Averin and K. Likharev, “Possible applications of the single charge tunneling,” in *Single Charge Tunneling—Coulomb Blockade Phenomena in Nanostructures*, ser. B, H. Grabert and M. Devoret, Eds. New York and London, U.K.: Plenum Press and NATO Scientific Affairs Division, 1992, vol. 294, NATO ASI, ch. 9, pp. 311–332.

[9] C. Lageweg, S. Cotofana, and S. Vassiliadis, “Static buffered set based logic gates,” in *Proc. 2nd IEEE Conf. Nanotechnology (NANO)*, 2002, pp. 491–494.

[10] C. Lageweg, S. Cotofana, and S. Vassiliadis, “Binary addition based on single electron tunneling devices,” in *Proc. 4th IEEE Conf. Nanotechnology (NANO)*, 2004, pp. 327–330.

[11] S. Cotofana, C. Lageweg, and S. Vassiliadis, “On computing addition related arithmetic operations via controlled transport of charge,” in *Proc. 16th IEEE Symp. Computer Arithmetic*, 2003, pp. 245–252.

[12] S. Cotofana, C. Lageweg, and S. Vassiliadis, “Addition related arithmetic operations via controlled transport of charge,” *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 243–256, Mar. 2005.

[13] C. Meenderinck and S. Cotofana, “High radix addition via conditional charge transport in single electron tunneling technology,” in *Proc. 16th IEEE Conf. Application-Specific Systems, Architectures and Processors (ASAP)*, 2005, pp. 294–299.

[14] C. Meenderinck, “Single electron technology based arithmetic operations,” Master’s thesis, Delft Univ. Technol., Delft, The Netherlands, 2005.

[15] C. Wasshuber, “About Single-Electron Devices and Circuits,” Ph.D., Tech. Univ. Vienna, Vienna, Austria, 1998.

[16] D. Averin and A. Odintsov, “Macroscopic quantum tunneling of the electric charge in small tunnel junctions,” *Phys. Lett. A*, vol. 140, no. 5, pp. 251–257, Sep. 1989.

[17] D. Averin and Y. Nazarov, “Virtual electron diffusion during quantum tunneling of the electric charge,” *Phys. Rev. Lett.*, vol. 65, no. 19, pp. 2446–2449, Nov. 1990.

[18] S. Lotkhov, H. Zangerle, A. Zorin, and J. Niemeyer, “Storage capabilities of a four-junction single-electron trap with an on-chip resistor,” *Appl. Phys. Lett.*, vol. 75, no. 17, pp. 2665–2667, Oct. 2006.

[19] 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. Appl. Phys.*, vol. 88, no. 5, pp. 2665–2670, Sep. 2000.

[20] S. Lotkhov, S. Bogoslovsky, A. Zorin, and J. Niemeyer, “Operation of a three-junction single-electron pump with on-chip resistors,” *Appl. Phys. Lett.*, vol. 78, no. 7, pp. 946–948, Feb. 2006.

[21] M. Sulieman and V. Beiu, “Design and analysis of SET circuits: Using MATLAB modules and SIMON,” in *Proc. 4th IEEE Conf. Nanotechnology*, 2004, pp. 618–621.

[22] K. Ishibashi, D. Tsuya, M. Suzuki, and Y. Aoyagi, “Fabrication of a single-electron inverter in multiwall carbon nanotubes,” *Appl. Phys. Lett.*, vol. 82, no. 19, pp. 3307–3309, Feb. 2001.

[23] C. Lageweg, S. Cotofana, and S. Vassiliadis, “A linear threshold gate implementation in single electron technology,” in *Proc. IEEE Computer Society Workshop VLSI*, 2001, pp. 93–98.

[24] SIMON: A single electron device and circuit simulator. [Online]. Available: http://www.lybrary.com/simon/

[25] H. Pothier, P. Lafarge, P. F. Orfila, C. Urbina, D. Esteve, and M. H. Devoret, “Single electron pump fabricated with ultrasmall normal tunnel junctions,” *Physica B*, vol. 169, pp. 573–574, Feb. 1991.

[26] M. Daglish and T. Kemmitt, “Ferroelectric thin films—Research, development and commercialisation,” *IPENZ Trans.*, vol. 27, no. 1, pp. 21–24, 2000.

[27] G. Subramanyam, F. Ahamed, R. Biggers, A. Campbell, R. Neidhard, E. Nykiel, R. Cortez, K. Stamper, and M. Calcatera, “A new ferroelectric varactor shunt switch for microwave and milimeterwave reconfigurable circuits,” *Frequenz*, vol. 59, no. 1-2, pp. 37–48, 2005.

[28] W. Clemens and D. Zipperer, “Organic capacitor having a voltage-controlled capacitance,” Int. Patent pCT/DE2004/001797, 2005.

Cor Meenderinck (S’05) received the M.Sc. degree in electrical engineering from the Delft University of Technology, The Netherlands. Currently, he is working toward the Ph.D. degree in the Computer Engineering Laboratory of the Delft University of Technology.

His research interests include nanoelectronics, single-electron tunneling, computer arithmetic, computer architecture, chip multiprocessors, media accelerators, design for variability, and design for power efficiency.

Sorin Cotofana (M’93–SM’00) received the M.S. degree in computer science from the Politehnica University of Bucharest, Romania, and the Ph.D. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands.

He worked for a decade with the Research and Development Institute for Electronic Components (ICCE), Bucharest. He is currently an Associate Professor in the Computer Engineering Laboratory of Delft University of Technology. His research interests include computer arithmetic, parallel architectures, embedded systems, nanotechnology, reconfigurable computing, computational geometry, and computer-aided design.

Dr. Cotofana is a member of the IEEE Computer Society.