Building Blocks for Electron Counting Arithmetic

Casper Lageweg, Sorin Cotofana and Stamatis Vassiliadis
Computer Engineering Lab
Faculty of Electrical Engineering, Mathematics and Computer Science Department
Delft University of Technology
Delft, The Netherlands
http://ce.et.tudelft.nl
E-mail: {Casper,Sorin,Stamatis}@CE.ET.TUDelft.NL

Abstract— **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 first briefly discuss the concept of electron counting based arithmetic. Second, we introduce the types of building blocks that are required in order to implement this concept in SET technology. These blocks can be divided in three function categories: encoding binary operands as quantities of charge, controlling charge transport, and re-converting quantities of charge to binary results. Finally, we propose possible SET based implementations of these building blocks, and demonstrate the designs by means of simulation.**

Keywords— **single electron technology, SET, electron counting arithmetic**

I. Introduction

Feature size reduction in microelectronic circuits has been an important contributing factor to the dramatic increase in the processing power of computer 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 [9]. Therefore, several emerging technologies are currently being investigated [5]. Single Electron Tunneling (SET) [2] 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 [8]) 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. 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 inhibits further charge transport, making it possible to control the transport of charge in discrete and accurate quantities.

The ability to control accurately the transport of individual electrons in SET technology introduces a broad range of new possibilities and challenges for implementing computer arithmetic circuits. One such possibility is the representation of numerical values by a discrete amount of electron charges, such that individual digits can assume more then 2 values (i.e., a non-binary number system). Such a representation would be most beneficial for (addition related) arithmetic operations, as it would reduce the carry chain (and hence potentially result in more compact solutions with reduced area, delay and power consumption) [4]. Resulting, the arithmetic operations can be performed directly in charge by controlling the transport of individual electrons. This methodology for implementing arithmetic operations will be referred to as *electron counting*.

Assuming that both the operands and the result of an arithmetic operations based on the electron counting paradigm must be represented in the classical binary form, we can identify that the utilization of the electron counting paradigm requires basic blocks implementing the following functionality:

1. The conversion of a binary number to a charge-encoded (non-binary) format.
2. The controlled transportation of electron charges as a function of (non-Boolean) input variables.
3. The conversion of a charge-encoded number into a binary format.

The implementation of these basic blocks was left open in our preliminary study of the electron counting paradigm [4]. In this paper, we propose possible SET based imple-

mentations of these building blocks, and demonstrate the designs by means of simulation.

The remainder of this paper is organized as follows. Section II briefly presents the SET background theory. Section III introduces the concept of electron counting based arithmetic and the types of building blocks required for implementing electron counting based schemes. In Section IV we propose implementations of these building blocks which are demonstrated by means of simulation. Finally, section V concludes the paper.

II. BACKGROUND

Single Electron Tunneling technology introduces the quantum tunnel junction as a new circuit element. A tunnel junction consist 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 then 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 [6], we calculate V_c for the junction as

$$V_c = \frac{e}{2(C_e + C_j)}. \tag{1}$$

In the equation above, as well as in the remainder of this discussion, we refer to the charge of the electron as $e = 1.602 \cdot 10^{-19}\ C$. Strictly speaking this is incorrect, as the charge of the electron is of course negative. However, it is more intuitive to consider the electron as a positive constant for the formulas which determine if a tunnel event takes place or not. We will of course correct for this when we discuss the direction in which the tunnel event takes place. Generally speaking, if we define the voltage across a junction as V_j, a tunnel event occurs 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.

Assuming that a tunnel event is possible, the orthodox theory for single electron tunneling (see for example [6] for a more extensive introduction) states that tunneling is a stochastic process, in which the rate at which tunnel events occur at $0K$ temperature is

$$\Gamma = \frac{|V_j - V_c|}{eR_t} \tag{2}$$

where R_t is the tunnel resistance (usually $\approx 10^5\Omega$). Note that a non-$0K$ temperature implies a lower event rate. Assuming that an individual tunnel event can be described as a Poisson process, we can calculate the required delay t for a single tunnel event to occur for a given error chance P_{err} as:

$$t = \frac{ln(P_{err})eR_t}{|V_j - V_c|} \tag{3}$$

Given that the minimum amount of transportable charge consists of a single electron, there exists a minimum energy threshold, called the Coulomb energy, which must be present in the circuit so that the transport of a single electron reduces the total amount of energy in the system. Resulting, in order to utilize the electron tunneling phenomenon, all other types of energy must be much smaller then the Coulomb energy. For thermal energy, this implies that, if we intend to add or remove charge to a circuit node by means of tunnel events, the total capacitance attached to such circuit nodes must be less then $900aF$ for $1K$ temperature operation, or less then $3aF$ for $300K$ (room temperature) operation [3]. This represents a major SET fabrication technology hurdle as even for cryostat temperature operation very small circuit features are required to implement such small capacitors. Another major technology challenge comes from the fact that thus far all experimental circuits have displayed a random offset charge (random charge present on circuit nodes), which is assumed to be the result of trapped charge particles in the tunnel junctions themselves or in the substrate. This random charge results in a random additional voltage across tunnel junctions, which can cause errors in their switching behavior. At the same time there are indications [2] that the offset charge problem may reduce or even disappear entire for the nanometer-scale feature size circuits required for room temperature operations. Given this and the fact that in our investigation we focus on the efficient utilization of the

SET behavioral properties we ignore the aspects related to offset charge and its potential influence on SET based computational structures.

III. Electron Counting Building Blocks

Given that we can control the transport of individual electrons, we have the possibility of encoding integer values X directly as a net extra charge Xq_e (X can assume larger values then 1). Once integer values have been encoded as a number of electrons, we can perform arithmetic operations directly in electron charges. Such an approach is based on the transport of electron charges under the control of input operands. This reveals a broad range of novel computational schemes, which we generally refer to as electron counting. A preliminary investigation [4] revealed that such an approach results in extremely compact schemes for implementing addition related operations.

When examining an arithmetic operation implemented in accordance with the electron counting paradigm, we assume the following. Electron counting circuitry will most likely serve as specialized hardware alongside conventional digital circuitry. As such it can be assumed that the source operands of electron counting based arithmetic operations are supplied in a binary format. As stated earlier, and electron counting encoded operand with value X is encoded as a net charge of Xq_e. Therefore, an n-bit binary operand X can be transformed to charge encoded format by a circuit computing $Xq_e = \Sigma_{i=0}^{n-1} x_i 2^i q_e$. In other words, we require a Digital to Analog Conversion (DAC).

Once operands are converted to a charge encoded format as described above, addition and addition related operations such as multiplication can be implemented using a single type of building block [4], which we refer to as an $MVke$ block. The $MVke$ block computes the product Vkq_e in the form of a charge encoded vale, where V is a charge encoded input value and k is an integer constant.

Finally, it can also be assumed that charge encoded results must also be converted back to a conventional binary output $y_i, i = 0, 1, \ldots, n-1$,. In other words, we require an Analog to Digital Conversion (ADC). One possible implementation of a SET based ADC that converts a charge encoded value Xq_e to a binary format is based on PSF building blocks. A PSF block implements a periodic symmetric function and each bit y_i can be described by a periodic symmetric function with period 2^{i+1} [4]. Thus each output bit y_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.

The implementation of the building blocks described above was left open in our preliminary study of the electron counting paradigm [4]. In the next section we propose possible SET based implementations of these building blocks.

IV. Possible Building Block Implementations

In this section we propose possible SET based implementations of the building blocks introduced in the previous section. We briefly discuss the operation principle of the proposed implementations, and demonstrate the behavior of the designs by means of simulation.

A. MT Building Block

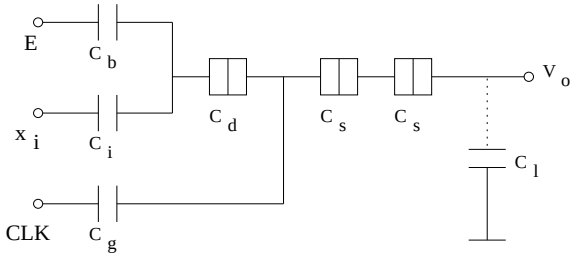

Fig. 1. Modified Turnstile (MT) building block.

An n-bit Digital to Analog Converter (DAC) is a circuit with binary inputs x_i, $i = 0, 1, \ldots, n-1$, and an analog output V_o. The DAC circuit can be utilized to convert an n-bit binary input X into its charge encoded equivalent by calculating an analog voltage V_o as $V_o = \Sigma_{i=0}^{n-1} x_i 2^i q_e / C_\Sigma$, where C_Σ is the total capacitance of the circuit's output node. A possible implementation of a SET DAC is based the Modified Turnstile (MT) building blocks [1] as depicted in Figure 1. When enabled and triggered by a clock signal CLK, the MT block adds a charge of q_e to its output node if the input $x_i = 1$. An input bit x_i in position i has to contribute to the output voltage with $x_i 2^i q_e / C_\Sigma$. Thus an n-bit DAC can be implemented by multiple MT blocks sharing a single output node, such that an input bit x_i drive 2^i MT blocks in parallel as depicted in Figure 2.

As an example of the DAC scheme, we present a 3-bit DAC, consisting of 7 MT blocks. The SIMON [7] simulation results are displayed in Figure 3. In this figure, starting from the top, the first two rows represent the enable and clock control signals. The third, fourth, and fifth row represent the binary inputs $Data_0$, $Data_1$, and $Data_2$. The bottom row represents the circuit output. As can be observed, due to the discrete nature of the charge stored the circuit's output the output values are exact multitudes of q_e/C_Σ, demonstrating that the MT block based conversion process functions as desired.

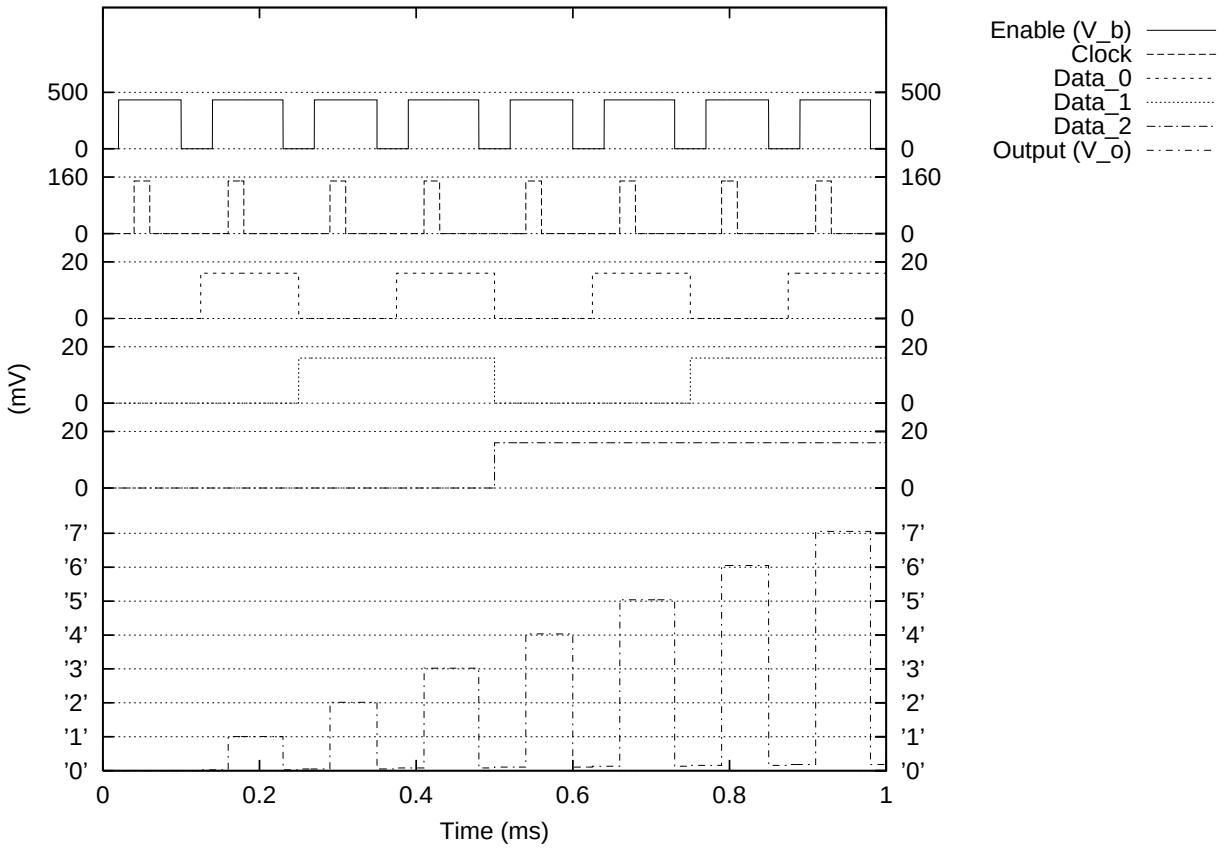

Fig. 3. Simulation results for the 3-bit DAC.

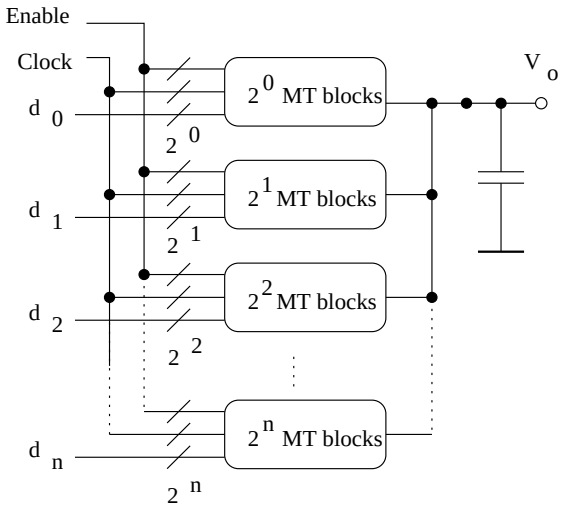

Fig. 2. Implementation of an n-bit DAC.

B. MVke Building Block

Preliminary investigations revealed that electron counting based addition and addition related operations such as multiplication can be implemented using a single type of building block [4], which we refer to as an $MVke$ block. When enabled and triggered by a clock signal CLK, the $MVke$ block transports a net charge Vkq_e to its output node Y, where V is a charge encoded operand and k an integer constant. Note that for $V = 1, k = 1$, the function of the $MVke$ building block corresponds with that of the MT blocks discussed above in Section IV-A. Thus, the $MVke$ block can also be utilized as a building block for a DAC as depicted in Figure 2.

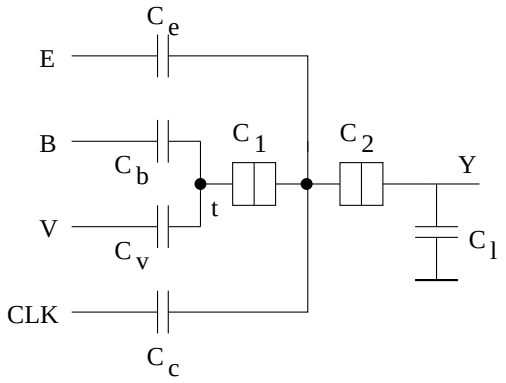

Fig. 4. $MVke$ block implementation.

A possible implementation of the $MVke$ block is dis-

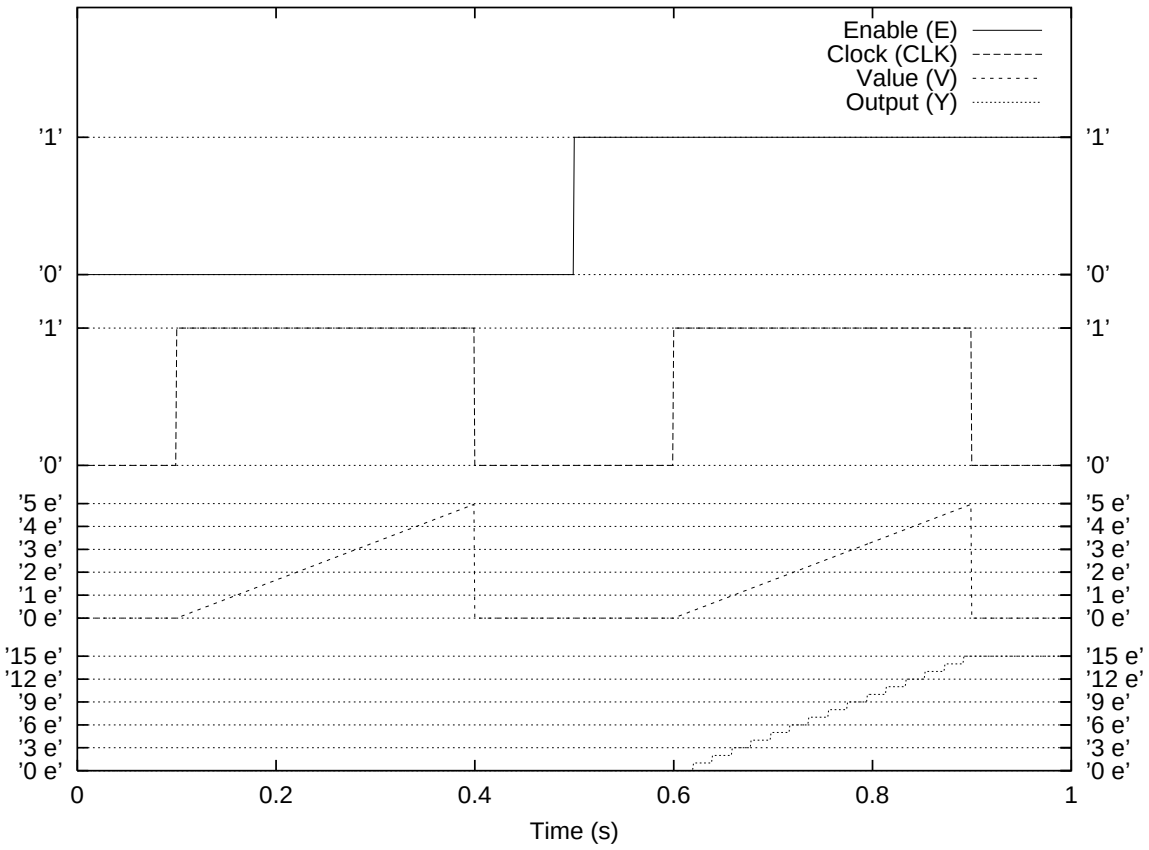

Fig. 5. $MVke$ block simulation results.

played in Figure 4. The circuit operates as follows. If a clock pulse CLK arrives, the SET transistor (C_1 and $C-2$) is opened if and only if $E = 1$. When the transistor opens, $V \times k \times q_e$ charge is added to the load capacitor C_l. As a result of this charge transport, an opposite charge $-V \times k \times e$ is stored on node 't'. The voltage resulting from this opposite charge cancels the effect of voltage source V, inhibiting further charge transport. The circuit is biased via the the DC input B. Given that the capacitor C_v acts as a weight factor for V, the desired multiplication constant value k can be adjusted by changing the value of C_k.

In order to demonstrate the proposed implementation of the $MVke$ block, as depicted in Figure 4, we have simulated an instance of the circuit with SIMON. In this example we set k, the multiplication factor, to $k = 3$. The simulation results of the block are depicted in Figure 5. In this figure, starting from the top, the first two rows represent the enable (E) and clock (E) inputs. The third row represents the input value Vq_e. The fourth row represents the output value Y. It can be observed that the circuit behavior is as described above, such that the output Y only responds to an input value V when the circuit is enabled and while a clock pulse is present. Also, one can observe that the output Y is indeed $Y = 3Vq_e$.

C. PSF Building Block

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 with the first positive transition and the first negative transition, as displayed in Figure 6. 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.

A possible implementation of the PSF block is displayed in Figure 4. The circuit operates as follows. The capacitor C_t and junction J_t form an electron trap structure. The charge encoded input value V serves as the input to the electron trap. Given than an electron trap circuit has a period output behavior, the electron trap's output node T has a periodic response to input V. The voltage on node T is capacitively added to a biasing voltage B and then serves as input for a SET inverter. The SET inverter behaves as a

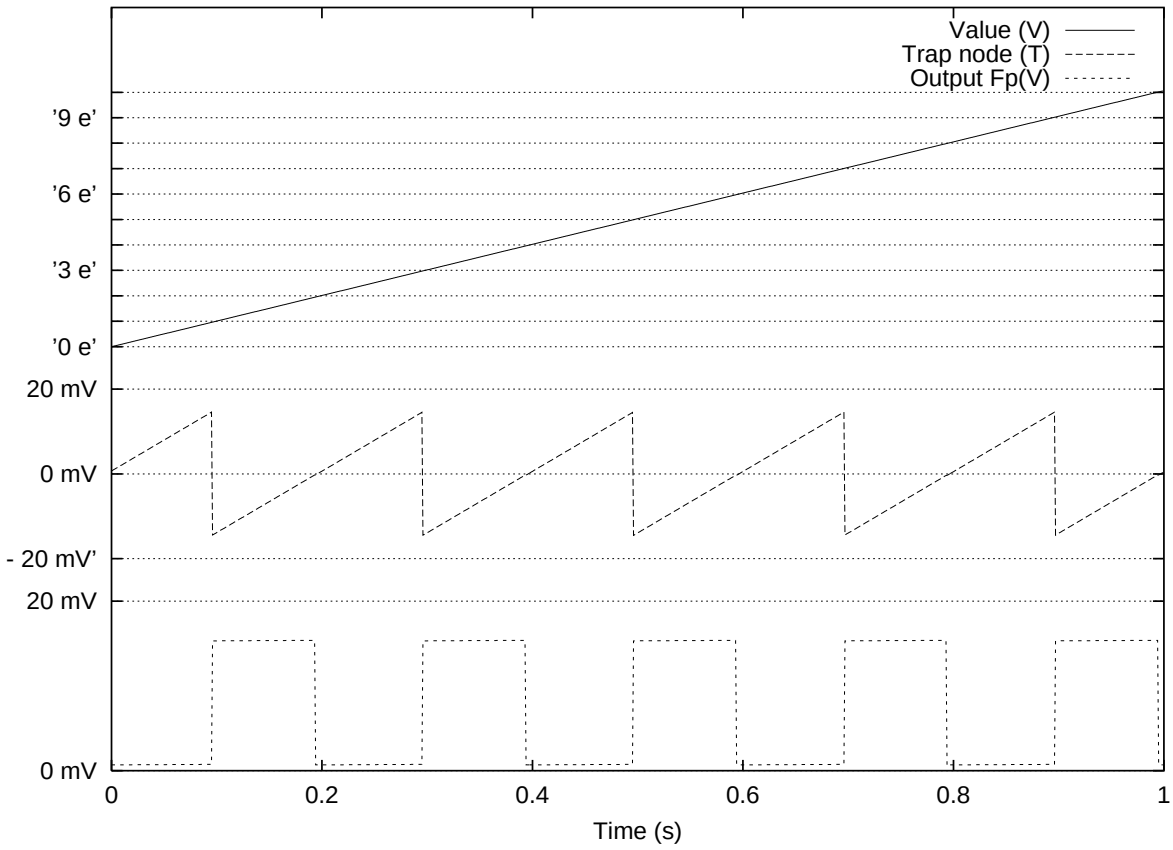

Fig. 8. PSF block simulation results.

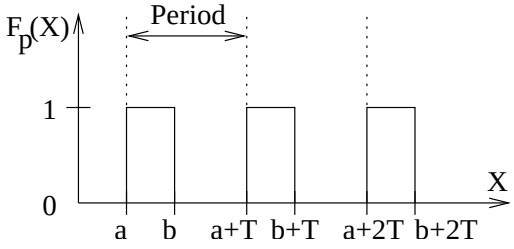

Fig. 6. Period symmetric function $F_p(X)$.

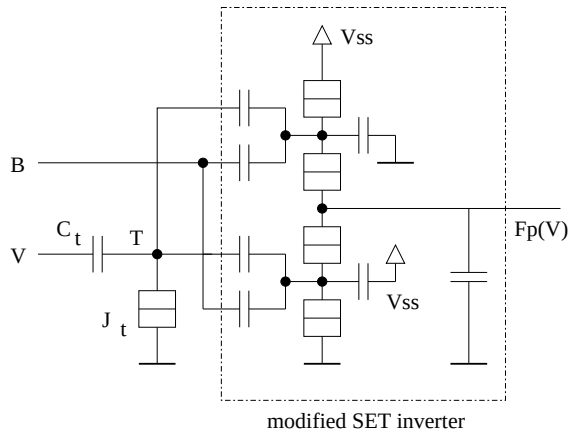

Fig. 7. PSF block implementation.

literal gate and transforms its input signal (within a limited range) to either logic 0 of logic 1.

In order to demonstrate the proposed implementation of the PSF block, as depicted in Figure 7, we have simulated an instance of the circuit with SIMON. In this example we set T, the period of the PSF gate, to $T = 2$. This results in $F_p(0) = F_p(2) = F_p(6)$ and $F_p(1) = F_p(3) = F_p(5)$, which corresponds with a parity function when $F_p(0) = 0$ and $F_p(1) = 1$. The simulation results of the block are depicted in Figure 8. In this figure, starting from the top, the first row represent the input value V. The seconds row displays the voltage present on node T. The last row represents the binary output $Fp(V)$. As can be observed, the gate correctly implements a parity function. Note that the period of the PSF gate can be adjusted via the electron trap's circuit parameters. Likewise, the biasing voltage B can be utilized to adjust which input values results in logic 0 and which in logic 1.

V. Conclusions

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 first briefly discussed the concept of electron counting based arithmetic. Second, we introduced the types of building blocks that are required in order to implement this concept in SET technology. These blocks can be divided in three function categories: encoding binary operands as quantities of charge, controlling charge transport, and re-converting quantities of charge to binary results. Finally, we proposed possible SET based implementations of these building blocks, and demonstrated the designs by means of simulation.

REFERENCES

[1] C.Lageweg and S.Cotofana and S.Vassiliadis. Digital to Analog Conversion Performed in Single Electron Technology. In *1st IEEE Conference on Nanotechnology (NANO)*, October 2001.

[2] K. Likharev. Single-Electron Devices and Their Applications. *Proceeding of the IEEE*, 87(4):606–632, April 1999.

[3] M.Goossens. *Analog Neural Networks in Single-Electron Tunneling Technology*. PhD thesis, Delft University of Technology, 1998.

[4] S. Cotofana and C. Lageweg and S. Vassiliadis. On Computing Addition Related Operations via Controlled Transport of Charge. In *16th IEEE Symposium on Computer Arithmetic (ARITH)*, pages 245–252, June 2003.

[5] Technology roadmap for nanoelectronics. Downloadable from website 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 Society Technologies (IST) program on Future and Emerging Technologies.

[6] C. Wasshuber. *About Single-Electron Devices and Circuits*. PhD thesis, TU Vienna, 1998.

[7] C. Wasshuber, H. Kosina, and S. Selberherr. SIMON - A Simulator for Single-Electron Tunnel Devices and Circuits. *IEEE Transactions on Computer-Aided Design*, 16(9):937–944, September 1997.

[8] 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 Transactions on Electron Devices*, Vol. 49(No. 3):pp. 193 –207, March 2000.

[9] Y.Taur, D.A.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. *Proceeding of the IEEE*, 85(4):486–504, 1997.