Computing Division in the Electron Counting Paradigm using Single Electron Tunneling Technology

Cor Meenderinck Sorin Cotofana
Computer Engineering Lab, Delft University of Technology, Delft, The Netherlands
{cor, sorin@ce.et.tudelft.nl}

***Abstract*— Single Electron Tunneling (SET) technology appears to be a promising alternative for CMOS as it exhibits excellent power consumption and scalability features. Furthermore, it allows to perform computation in unconventional, but efficient ways which is exploited by the Electron Counting (EC) paradigm. In this paper we investigate SET division schemes that operate under the EC paradigm. We propose two schemes of which the fastest has a linear time complexity. The implementation of the latter is discussed in detail and simulation results are presented. The worst case delay is calculated as 39.2 ns and the worst case energy consumption is calculated as 1.23 eV. The area cost of the scheme is 146 elements.**

***Index Terms*— single electron tunneling, division, computer arithmetic**

I. Introduction

Expected limitations to the scalability of existing semiconductor technologies, i.e., CMOS, necessitates the investigation of alternative nanoelectronics technologies. A promising candidate is Single Electron Tunneling (SET) technology, which has the potential of performing computation with lower power consumption than CMOS and is scalable to the nanometer region and beyond [1].

Several proposals have been made to implement computational operations using SET technology and these implementations are mainly categorized in two types. The first type of implementation represents logic values by voltage (see [2] for an overview) while the second type of implementation represents bits by single electrons.

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. The Electron Counting (EC) paradigm [3], on the other hand, uses a novel design style and appears promising as an efficient methodology for the implementation of SET based arithmetic operations. In previous research [4] addition related arithmetic operations, i.e., addition and multiplication, have been implemented utilizing the EC paradigm. In this paper we investigate the implementation of EC based division. First, using previously designed EC building blocks, we implement a basic division scheme. Subsequently, to improve circuit delay, we present an improved division scheme and explain the implementation in detail.

The remainder of this paper is organized as follows. Section II briefly describes the single electron tunnel phenomenon and introduces the EC paradigm. Section III introduces a basic EC division scheme that is based on a feedback loop. In Section IV a modification to the basic division scheme, resulting in smaller computation time, is proposed and simulation results of this scheme are presented. Section V concludes the paper and presents the future work.

II. Background

SET circuits are based on tunnel junctions which consist of an ultra-thin insulating layer in a conducting material (see Figure 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 $V_c = \frac{q_e}{2(C_e+C_j)}$ [5], 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 $|V_j| \geq V_c$.

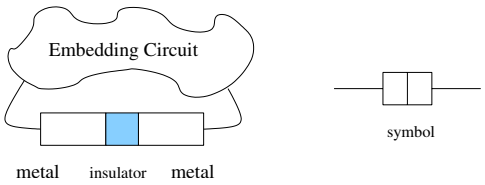

Fig. 1. Schematic representation of the tunnel junction.

Electron tunneling is stochastic in nature and as such the delay cannot be analyzed in the traditional sense. Instead, for each transported electron one can describe the switching delay as $t_d = \frac{-ln(P_{error})q_eR_t}{|V_j|-V_c}$, where R_t is the junction's resistance and P_{error} is the chance that the desired charge transport has not occurred after t_d seconds. In this paper we assume $R_t = 10^5\Omega$ and $P_{error} = 10^{-8}$. Each transported electron reduces the system energy by $\Delta E = q_e(|V_j| - V_c)$ from which the consumed energy can be calculated.

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 [6]. Therefore, circuit area is evaluated in terms the total number of circuit elements (capacitors and junctions).

There are many ways to do computation using SET technology, but most do not seem to fully utilize potential offered by SET. The Electron Counting (EC) paradigm is a novel way of computation that exploits 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 integer values X directly as a charge Xq_e . Using a Digital to Analog Converter (DAC) [3] 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, one can perform arithmetic operations directly in electron charges, which reveals a broad range of novel computational schemes [3], [4]. The result of the arithmetic operations can be converted back to a binary value using an Analog to Digital Converter (ADC) [3].

III. Basic EC division

Binary division is defined as $z = (q\mathrm{x}d) + s$ 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.

A basic way to make an EC based division scheme is presented in Figure 2 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 [3], which computes the product of d and the quotient q, and stores this value in charge reservoir $Q\mathrm{x}D$. 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 $Q\mathrm{x}D$ and allows a current source to subtract electrons from reservoir Q as long as $Q\mathrm{x}D$ is smaller than Z. In our implementations of EC based circuits we represent values by positive charge, meaning that removing electrons from a charge reservoir increases the represented value. When the value in reservoir $Q\mathrm{x}D$ 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 non integer quotients this scheme always rounds off upward.

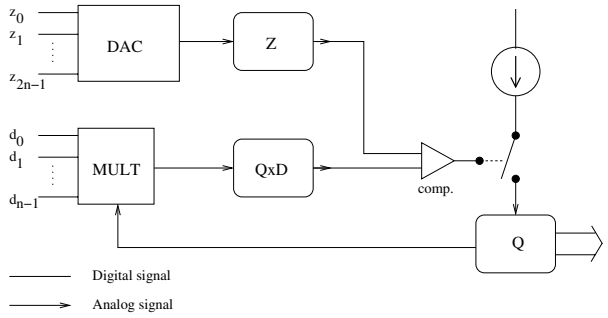

Fig. 2. Basic EC division scheme.

The division scheme utilizes a DAC building block of which a 4-bit instance is depicted in Figure 3. An n-bit DAC block contains n $MVke$ blocks. An $MVke$ block has inputs V, $enable$ and $reset$ and can remove Vk (k is a build in constant) electrons from the output reservoir when enabled. To do the conversion each input bit d_i is connected to the V input of an $MVke$ block with $k = 2^i$. Consequently, the total number of electrons removed from the charge reservoir is $\sum_{i=0}^{n-1} d_i 2^i$, which is the analog equivalent of the binary input.

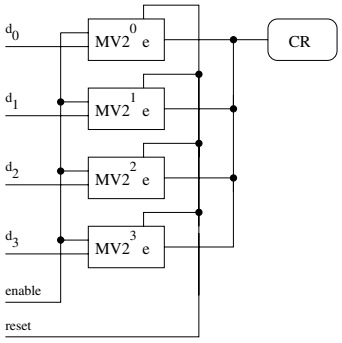

Fig. 3. 4 bit DAC scheme

The division scheme also utilizes an EC multiplier of which a 2-bit instance is depicted in Figure 4 and works as follows. First, input B is converted into analog and stored in a charge reservoir, using a DAC scheme as presented above. Subsequently, the analog value of B is fed into the V input of the upper $MVke$ blocks, while the bits of input A are connected to the $enable$ inputs. Consequently, the total number of electrons removed from the output reservoir is $\sum_{i=0}^{n-1} Ba_i 2^i = BA$.

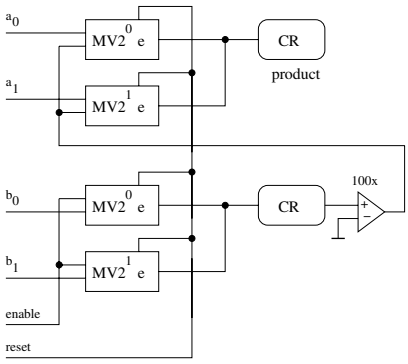

Fig. 4. 2 bit multiplication scheme

We implemented the division scheme of Figure 2 in the simulation environment SIMON [7]. The comparator was implemented using a SET Threshold Logic Gate (TLG) [8]. The current source and switch were implemented by a single electron transistor (see [9] for an early analysis), which operates similar to a MOS transistor. Though, 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. However, due to this characteristic of electron tunneling, the delay of the operation is rather large.

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 adding a delay in the feedback loop. This could be implemented by replacing the single electron transistor with an electron pump [10], but the delay would worsen even more.

The basic division scheme as presented so far, does not compute the remainder. In order to do so, we could extend the circuit as follows. We define a tentative remainder $tr = Z - Q\text{x}D$, which can be computed as the difference from reservoir Z and reservoir $Q\text{x}D$. Note that $tr \leq 0$, due to the fact that this scheme always rounds off upward for q. The true remainder and quotient can be computed respectively as $s = tr + d$ and $q = Q - 1$, for $tr \neq 0$. To implement this we need a subtraction block, an addition block and a 'move conditional' block ($MCke$) [4]. The first one is currently under investigation, while the others are available as they were already implemented.

Although both adjustments could be implemented, we actually didn't do it because of the associated delay and hardware costs. Instead, we investigated another scheme that we present in the next section.

IV. Improved EC 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.

A. Strategy

The analog value of Q can be described as a sum of powers of 2, 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 calculate Q.

The EC division scheme as depicted in Figure 5 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 block are consecutively enabled, starting with the most significant one. $MCke$ block i ($i = n-1, n-2, .., 0$) removes $k = 2^i$ electrons from reservoir Q if the condition $\Psi_i = (Z - Q\text{x}D > 2^i D)$ 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 $Q\text{x}D$) that is smaller or larger than Z. 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 non integer quotients this scheme rounds off downward.

The remainder, defined as $s = z - q\text{x}d$, can easily be computed in this scheme using a subtraction block. The subtraction block uses as inputs the analog values from reservoirs Z and $Q\text{x}D$ 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., $Q\text{x}D$, is 0. The difference between Z and its estimation is 14, which is smaller than $2^i D$. 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 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.

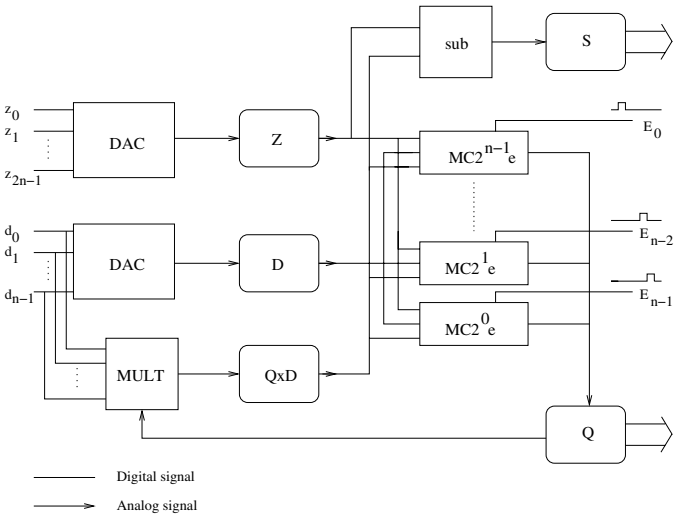

Fig. 5. Improved EC division scheme.

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

Q	DQ	$Z - DQ$	S	i	$D2^i$	$Z - DQ - 2^i D$	Ψ
0	0	14	14	2	16	-2	FALSE
0	0	14	14	1	8	6	TRUE
2	8	6	6	0	4	2	TRUE
3			2				

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 which is n for a $2n$-bit by n-bit divider. Therefore in general the delay is linear to the number of input bits.

B. Implementation

The improved EC division scheme requires some extra hardware compared to the basic version. One of the extra building blocks required is the $MCke$ block of which an implementation was proposed in [4]. The $MCke$ building block consists of two parts, a 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. Since the original 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 of Figure 6.

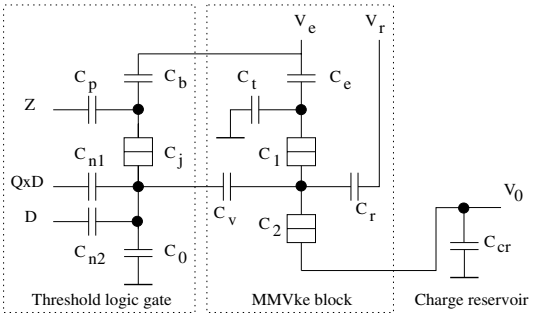

Fig. 6. Modified $MCke$ building block.

A threshold gate is a device that computes a Boolean function given by:

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

$$\mathcal{F}(X) = \sum_{i=1}^{n} \omega_i x_i - \phi \quad (2)$$

where x_i are the Boolean inputs, w_i are the corresponding weights, and ϕ is an internal threshold value. Condition Ψ_i can be expressed as $\mathcal{F}(X) = Z - Q\text{x}D - 2^i D$, thus it can be implemented by a three input threshold gate, having $X = \{Z, Q\text{x}D, D\}$, $W = \{1, -1, -2^i\}$ and $\phi = 0$, as depicted in Figure 6. For more details on the implementation of the SET threshold gate we refer the reader to [8]. The implementation of the $MMVke$ block is discussed in detail in [4] .

The linear division scheme requires a set of enable signals, which together have the shape of a pulse train. Preferably these signals are generated by local control logic, which we did not implement so far, but intent to using a delay line structure, build out of serially cascaded SET inverters [11]. This would even allow us to assign to each $MCke$ block a different delay time, depending on the number of electrons it has to transfer.

The subtraction block is currently under investigation. We are researching an implementation using $MVke$ blocks that can add electrons to a reservoir, as opposed to the current implementation of the $MVke$ block which can only remove electrons.

To verify the proposed linear EC based division scheme we simulated a 6-bit by 3-bit divider (disregarding the remainder computation). Figure 7 presents the simulation results for $14/3$ and $14/4$ of which the first one corresponds to the numbers in Table I. The top two rows show respectively the reset and (block level) enable signal. The next row shows the value of $Z - Q\text{x}D$, which is actually equal to the value of the remainder. The next three rows are the (internal) enable signals for the $MCke$ blocks and the last row represents the quotient.

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 39.2 ns and the worst case energy consumption was calculated as 1.23 eV. The area cost of the scheme is 146 elements.

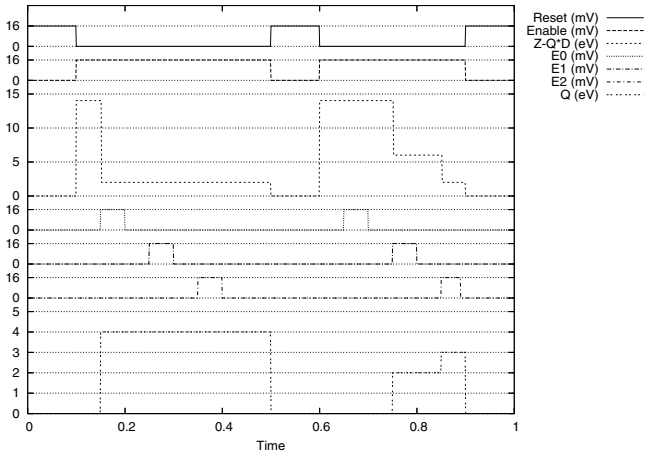

Fig. 7. Simulation results for 14/3 and 14/4.

V. Conclusions and Future Work

In this paper we presented two EC based division schemes in SET technology. The first and basic scheme suffers from overshoot and has a rather large delay. The second scheme, which is a modification of the basic scheme, has linear time complexity and all simulations indicate that this scheme works correctly. We calculated the worst case delay of this scheme as 39.2 ns and the worst case energy consumption as 1.23 eV.

The second division scheme requires control logic of which no implementation was presented. We are currently investigating a possible implementation which was briefly described in this paper. Also we currently working on the implementation of the subtraction block. Further, we plan to investigate a division scheme based on periodic symmetric functions which could potentially compute division in $O(1)$ time.

References

[1] "International Technology Roadmap for Semiconductors, 2003 Edition, Executive Summary," Downloadable from website http://public.itrs.net/Home.htm, 2003.

[2] K. Likharev, "Single-Electron Devices and Their Applications," *Proceeding of the IEEE*, vol. 87, no. 4, pp. 606–632, April 1999.

[3] S. Cotofana, C. Lageweg, and S. Vassiliadis, "Addition Related Arithmetic Operations via Controlled Transport of Charge," *IEEE Transactions of Computers*, vol. 54, no. 3, pp. 243–256, March 2005.

[4] C. Meenderinck, "Single electron technology based arithmetic operations," 2005, Master's thesis, CE-MS-2005-01.

[5] C. Wasshuber, "About single-electron devices and circuits," Ph.D. dissertation, TU Vienna, 1998.

[6] 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, February 2001.

[7] http://www.lybrary.com/simon/.

[8] C. Lageweg, S. Cotofana, and S. Vassiliadis, "A Linear Threshold Gate Implementation in Single Electron Technology," in *IEEE Computer Society Workshop on VLSI*, April 2001, pp. 93–98.

[9] K. Likharev, "Single-Electron Transistors: Electronic Analogs of the DC Squids," *IEEE Transactions on Magnetics*, vol. MG-23, pp. 1142–1145, March 1987.

[10] 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, February 1991.

[11] C. R. Lageweg, S. D. Cotofana, and S. Vassiliadis, "Static buffered set based logic gates," in *Proceedings of the 2002 2nd IEEE Conference on Nanotechnology*, August 2002, pp. 491–494.