High-Radix Addition and Multiplication in the Electron Counting Paradigm Using Single Electron Tunneling Technology

Cor Meenderinck and Sorin Cotofana

Delft University of Technology, Computer Engineering Lab
Postbus 5031, 2600 GA, Delft, The Netherlands
{cor, sorin}@ce.et.tudeft.nl

Abstract. The Electron Counting (EC) paradigm was proved to be an efficient methodology for computing arithmetic operations in Single Electron Tunneling (SET) technology. In previous research EC based addition and multiplication have been implemented. However, the effective performance of these schemes is diminished by fabrication technology imposed practical limitations. To alleviate this problem high radix computation was suggested. In this paper we present a high radix EC addition scheme and a high radix EC multiplication scheme. For both arithmetic operations, we first briefly present the normal (non high radix) EC schemes. Second, we present the high radix schemes and explain their functionality. Third, we explain the implementation of the high radix schemes in details. Finally, we present simulation results and evaluate the schemes in terms of delay and area cost.

1 Introduction

It is generally expected that current semiconductor technologies, i.e., CMOS, cannot be pushed beyond a certain limit because of problems arising in the area of power consumption and scalability. A promising alternative is Single Electron Tunneling (SET) technology [1], which has the potential of performing computation with lower power consumption than CMOS and is scalable to the nanometer region and beyond [2].

Several proposals have been made to implement computational operations using SET technology and these implementations are mainly categorized in two types (see for example [1,3]). The first type of implementation represents logic values by voltage (see [3] for an overview) while the second type of implementation represents bits by single electrons. Single Electron Encoded Logic (SEEL) [4] is an example of the latter.

Using the second type of implementation, arithmetic units can be designed in conventional logic design styles, e.g., using Boolean and/or threshold gates (see for example [4]). The Electron Counting (EC) paradigm [5], on the other hand, uses a novel design style and appears promising as an efficient computational paradigm for the implementation of SET based arithmetic operations, e.g., addition and multiplication. Previous EC based adder and multiplier implementations assumed that an unlimited amount of electrons could be transported within the EC building blocks, which does not hold true in practice. Therefore, a limit to the operand size of the previous proposed schemes

S. Vassiliadis et al. (Eds.): SAMOS 2006, LNCS 4017, pp. 447–456, 2006.

is implied by the available SET fabrication technology. One way to alleviate this problem is to do high-radix computation [6]. In this paper we propose a high-radix EC addition scheme and a high radix EC multiplication scheme.

The remainder of this paper is organized as follows. Section 2 briefly describes the single electron tunnel phenomenon and introduces the EC paradigm. Section 3 introduces the proposed high radix addition scheme, explains the implementation details and presents simulation results. In Section 4 the high radix multiplication scheme is proposed and explained in details. The scheme is verified by means of simulation. Section 5 concludes the paper.

2 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 [7] $V_c = \frac{q_e}{2(C_e+C_j)}$, 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 $|V_j| \geq V_c$.

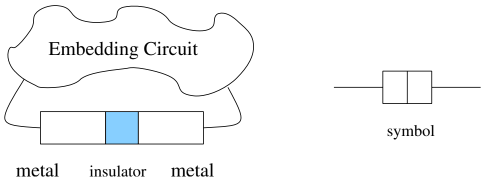

Fig. 1. Schematic representation of the tunnel junction

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 $P_{error} = 10^{-8}$.

Note that the implementations discussed in here are technology independent. SET tunnel junctions can for example be implemented by classical semiconductor lithography or by carbon nanotubes [8]. Therefore, circuit area is evaluated in terms the total number of circuit elements (capacitors and junctions).

As mentioned in the introduction, there are many ways to do computation using SET technology of which the Electron Counting paradigm seems to exploit the potential of SET most of all. 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. 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 High-Radix EC Adder

A basic (non high radix) EC based addition scheme was first proposed in [5]. Figure 2 depicts a 2-bit instance of this addition scheme, which functions as follows. Each bit of the inputs A and B is connected to the V input of an $MVke$ building block. This block, once enabled, adds Vkq_e charge to the charge reservoir, where V is the magnitude of the input, k is a build in constant, and q_e is the absolute charge of one electron $(1.602 * 10^{-19}\,\text{C})$. In other words, the $MVke$ block removes Vk electron from the charge reservoir. The build in constant k is adjusted to the weight of the corresponding input bit of the $MVke$ block. Thus, the total amount of electrons removed from the charge reservoir is $\sum_{i=0}^{1}(a_i 2^i + b_i 2^i)$, which is equal to the sum of the inputs. The results of the addition is converted back to the digital domain using three PSF building blocks, which each implementing a Periodic Symmetric Function (PSF).

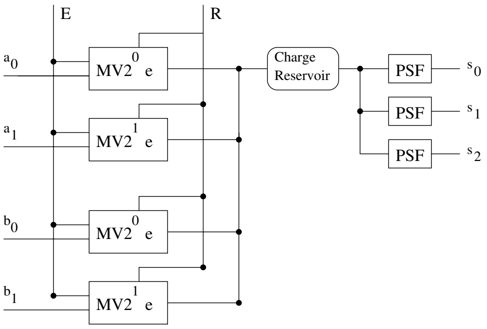

Fig. 2. 2-bit EC addition scheme

A possible implementation of the $MVke$ building block (Figure 3.1) was proposed in [6] and operates as follows. While R (reset) and V are zero and input E (enable) is set to '1', the voltage over junction 1 (C1) approaches it's critical voltage. If now V is set to '1' the critical voltage is exceeded and one electron tunnels from node M to node N. As a result of this event a positive charge is present on node M, which causes the voltage over junction 2 (C2) to exceed it's critical voltage and so one electron tunnels from node P to node M. This process of two tunnel events continues until the voltage over junction 1 has dropped below the critical voltage again. The number of electrons k that is removed from the charge reservoir is proportional to the magnitude of both V and C_v.

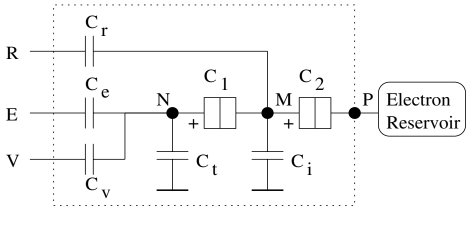

Fig. 3.1. *MVke* block implementation

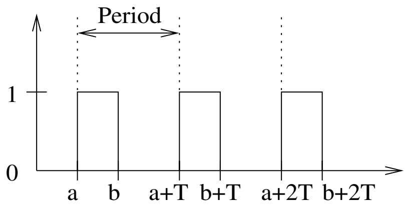

Fig. 3.2. Periodic Symmetric Function

A PSF block implements a Periodic Symmetric Function (PSF) F_s whose output is logic '1' within an interval from a to b, and with a period T (see Figure 3.2). Each bit s_i of a digital representation of a value X can be described as a PSF of X as $s_i = F_{s,i}(X)$, where the period is 2^{i+1}. Thus, utilizing a *PSF* block for each bit an analog to digital conversion can be performed.

An implementation of the *PSF* block is depicted in Figure 4.1 and was also proposed in [6]. The capacitor C_c and the tunnel junction C_t form an electron trap, which has a periodic transfer function. If the input voltage rises, the output voltage follows, due to capacitance division. At some point, though, the voltage over the tunnel junction exceeds the critical voltage and an electron tunnels to node T. The voltage of node T drops therefore. As the input voltage continues to rise, the voltage of node T rises again until it reaches the critical voltage. To obtain a *PSF* implementation a SET inverter [4] was added, which functions as a literal gate. As long as the input is below the threshold value, the output is ’0’, if the input exceeds the threshold, the output value becomes ’1’.

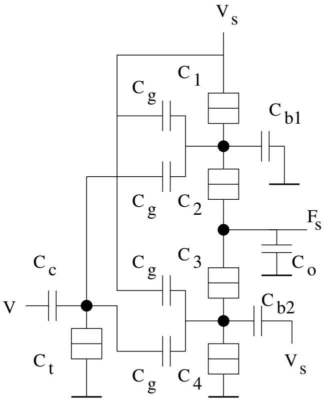

Fig. 4.1. *PSF* block implementation

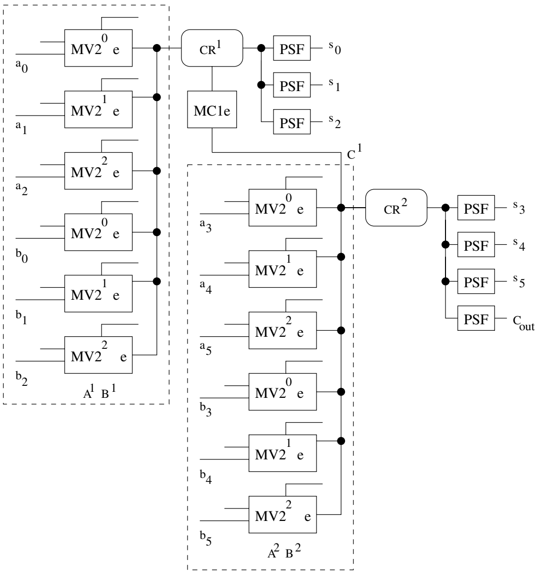

Fig. 4.2. Organization of the 6-bit radix-8 adder

3.1 High Radix Strategy

In order to create a high radix EC addition scheme only small adjustments to the normal EC addition scheme are required. Figure 4.2 depicts a 6-bit radix-8 adder which operates as follows. Each 6-bit input is split into parts of three bits. The lower bits are added and stored in CR^1 using a normal 3-bit EC adder, while the higher bits are added and stored in CR^2. Consequently, charge reservoirs CR^1 and CR^2 contain the intermediate sums $IS^1 = \Sigma_{i=0}^{2}(a_i 2^i + b_i 2^i)q_e$ and $IS^2 = \Sigma_{i=3}^{5}(a_i 2^{i-3} + b_i 2^{i-3})q_e$, respectively. If the intermediate sum $IS^1 > 7q_e$, a carry signal is generated by the $MC1e$ block, adding $1q_e$ charge to CR^2, thus *conditionally* moving $1q_e$ charge. Finally, the charge values present in CR^1 and CR^2 are each converted to a binary representation by means of three and four *PSF* blocks, respectively.

3.2 Implementation

While the implementations of the *MVke* and the *PSF* block were proposed in [6] no implementation of the *MC*1*e* building block was previously introduced. This section presents a possible implementation of a generalized version of the *MC*1*e* block, i.e., the *MCke* block, that moves kq_e charge into a charge reservoir if its input value exceeds a certain value. Such a block constitutes a generalization of the *MC*1*e* block and it is also useful for a wider range of EC based arithmetic operations.

The *MCke* block has two Boolean inputs (enable V_e and reset V_r), one analog input V_v, and one output connected to a charge reservoir (see Figure 5). Note that the voltage V_v can be either a voltage source or the value of a charge reservoir. If $V_e =$'1', $V_r =$'0', and V_v exceeds a threshold ψ, then the block removes k electrons from the charge reservoir connected to the output. If $V_r =$'1' (reset) and $V_e =$'0' all the electrons which were previously removed from the reservoir are returned.

The functionality implemented by an *MCke* block can be thought of as consisting of two stages. The first stage detects whether V_v exceeds the threshold value ψ. If $V_v > \psi$, the Boolean output Y of the first stage is set to '1', otherwise it is set to '0'. Consequently, the operation performed by the first stage of the *MCke* block can be described as an 1-input threshold function. The SET threshold gate proposed in [9] can be utilized to realize the first stage of the *MCke* block.

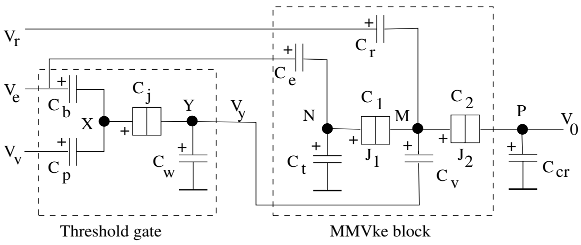

Fig. 5. *MCke* block implementation

The second stage of the *MCke* block removes k electrons from a charge reservoir when $V_r =$'0' and $V_e = V_y =$'1'. The *MVke* block proposed in [6] performs a similar function and can be adjusted to operate as specified above. The resulting Modified Move k electrons (*MMVke*) block functions slightly different than the *MVke* block. The enable signal V_e is used to set a positive voltage over junction 2 (J2) and junction 1 (J1). Capacitance values are such that the voltage over junction 2 is close to its critical voltage while the voltage over junction 1 stays below its critical voltage. The driving input V_y is connected through a capacitor to the central node M. If $V_y =$'0' no tunnel event can take place, but if $V_y =$'1' the voltage over J2 exceeds its critical voltage and k electrons tunnel from node P to node N. For an extensive overview of the *MMVke* block the reader is referred to [10].

3.3 Simulation

We have verified the *MCke* implementation by means of simulation using the SET simulation package SIMON [11]. Simulation results were obtained using the following circuit parameters: logic '0'$= 0mV$, logic '1'$= 16mV$, and the capacitance of the charge reservoir is $10^{-14}F$, in correspondence with previously designed EC building blocks [6]. Assuming a threshold voltage of $12.2mV$, as needed for the 6-bit radix 8 adder, we calculated the following parameters for the threshold gate: $C_w = 10aF$, $C_j = 0.5aF$, $C_p = 5aF$ and $C_b = 7.4aF$. The voltage swing on the output of the threshold gate was calculated as $15mV$. For $k = 1$ the following parameters were derived for the $MMVke$ block: $C_p = 5aF$, $C_b = 5.25aF$, $C_j = 0.5aF$, $C_w = 50aF$, $C_v = 2aF$, $C_1 = 7.43aF$, $C_2 = 0.5aF$, $C_e = 200aF$, $C_t = 100aF$ and $C_r' = 5aF$. All simulations indicate that this building block functions correctly.

We also simulated the 6-bit radix 8 addition scheme as depicted in Figure 4.2. The parameters of the utilized $MVke$ and PSF building blocks are those used in the EC adder presented in [6]. As stated earlier, a carry of one electron must be generated by the $MCke$ block when the CR^1 reservoir contains eight or more electrons. Thus the threshold voltage of the $MCke$ block should be the voltage corresponding with a charge of 7.5 electrons on the capacitance of the CR^1 reservoir. Therefore, the threshold voltage V_{vth} was set to $12.2mV$, which corresponds with the threshold voltage used for the $MCke$ block in the simulation mentioned above.

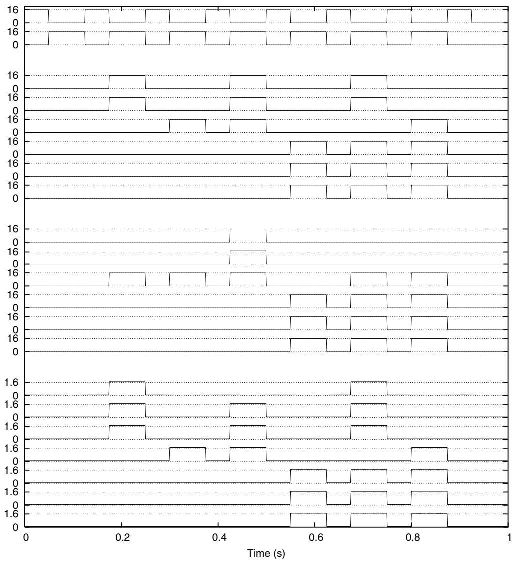

Fig. 6. Simulation results for the 6-bit radix-8 adder. From top to bottom (LSB first): reset(1), enable(1), input $A(6)$, input $B(6)$, sum(7)

The simulation results are depicted in Figure 6. The top block of two signals represent the reset and the enable, respectively. The second two blocks, each containing six signals, represent the input vectors A and B, respectively. The bottom block, containing seven signals represents the output vector of the adder. For each vector displayed in the graph, the top bar represents the least significant bit while the bottom bar represents the most significant bit. The test vectors A and B were chosen such that the $MCke$ block was tested for correct functionality under some extreme operating conditions. The simulation indicates that the high radix adder functions correctly. The adder requires 187 circuit elements and has a delay of 13.7 *ns*.

4 High-Radix EC Multiplier

Figure 7 depicts a basic (non high radix) multiplication scheme (first proposed in [5]), which operates as follows. Each bit of operand B is connected to an $MVke$ block, which adds $2^i q_e$ charge to the bottom charge reservoir if input b_i is logic '1'. Consequently, the charge reservoir contains the intermediate sum $IS = \Sigma_{i=0}^{1} b_i 2^i q_e$. This intermediate sum is fed into the V inputs of a next set of $MVke$ blocks. Therefore, these blocks add $IS*a_i*2^iq_e$ charge to the top charge reservoir. Consequently, that reservoir contains the product of inputs A and B. The value in the top reservoir is analog and can be converted to the digital domain using PSF blocks as it was done in the EC addition scheme (Figure 2). The multiplication scheme contains an OpAmp which has not been designed yet, but which can potentially be implemented using a hybrid FET-SET technology [12].

4.1 High Radix Strategy

The high-radix EC multiplication scheme we propose is based on the full-tree multiplication strategy [13] often used in fast multipliers, which comprises three steps. In the first step all partial products are produced at once in parallel. When assuming binary operands this can be done by simple AND-gates and for n-bit operands, this first step produces n rows of bits. In the second step, the number of rows is reduced using one or more stages of counters. With each stage of counters, the number of rows is reduced until only two rows are left over. In the last step these two bit rows are added, often by using a fast addition scheme like carry look-ahead.

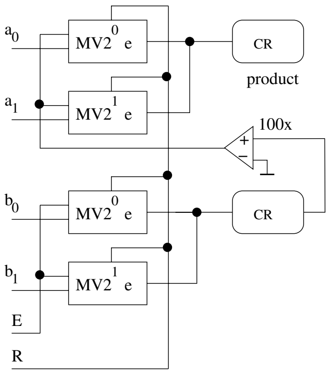

Fig. 7. 2-bit EC multiplication scheme

The strategy of the high-radix EC multiplication we propose, which is depicted in Figure 9.1 for the case of 8-bit radix 4 multiplication, comprises the same three steps with some adjustments. In the next section each step is explained in more detail.

4.2 Implementation

In the *first step*, the partial products are formed. Since we work in radix r, the operands are split into digits of log_2r bits. Assuming an operand size of n bits, this results in $\lceil \frac{n}{log_2r} \rceil$ digits for each operand. The multiplication of these digits is performed by normal EC multipliers, of which a 2-bit instance is depicted in Figure 7.

The direct application of the EC multiplication scheme for step one requires $(\frac{n}{log_2r})^2$ such multipliers. However, we can reduce the number of elements if we observe that each digit of B can be converted to analog once, after which this analog value is used by all multipliers in the same row (see Figure 8).

In the *second step* the number of rows is reduced by EC counters, which functionality is similar to normal population counters [14] used for binary operands. However, an EC counter assumes a number (k) of analog high-radix (r) inputs, all having the same weight, i.e., the inputs are all in the same column. The EC counter produces a number of outputs (s) in the same radix as the inputs, representing the sum of the inputs values, i.e., it produces a row. In the remainder of this paper we denote a specific instance of such an EC counter as EC (k,r;s) counter.

An EC counter implementation is depicted in Figure 9.2 for the case of four radix 16 inputs and operates as follows. Each analog input is buffered and amplified before

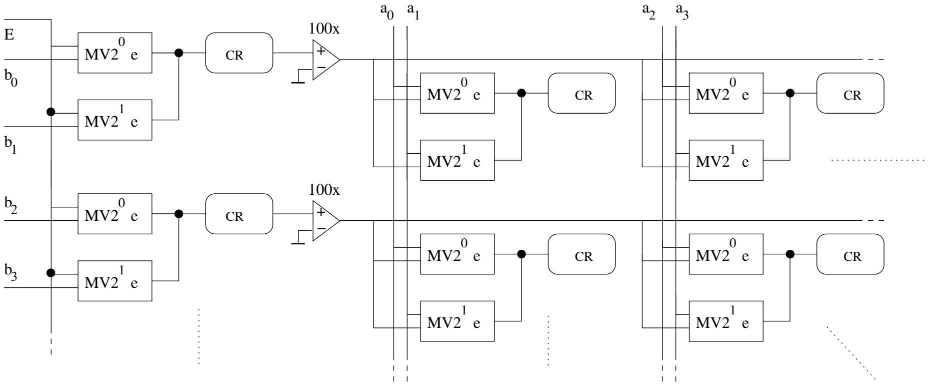

Fig. 8. Step one of high-radix EC multiplication

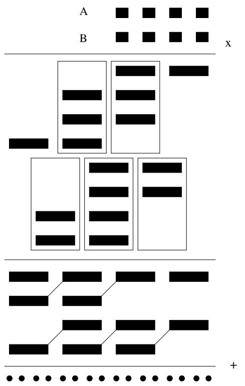

Fig. 9.1. 8-bit radix 4 EC multiplication strategy

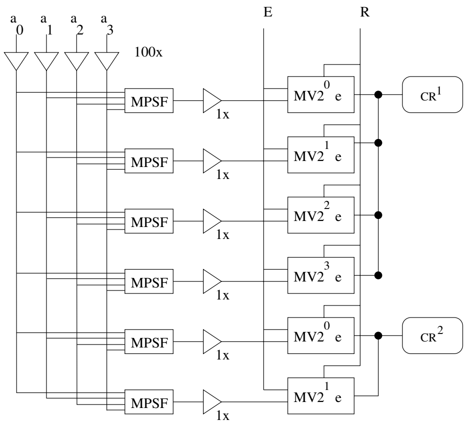

Fig. 9.2. EC (4,16;2) counter

it is fed into the *MPSF* blocks [15] in order to eliminate feedback effects. The *MPSF* blocks, which are multiple input versions of the *PSF* block depicted in Figure 4.1, perform both the addition of the inputs and the conversion of the intermediate sum to the digital domain. Once the intermediate sum is converted to the digital domain, it is split into digits of log_2r bits, in order to guaranty that the output is in the correct radix. These digits are each converted back to the analog domain by sets of *MVke* blocks and the final result is stored in charge reservoirs.

Since in our case the maximum intermediate sum is 64, six *MPSF* blocks are required each one producing one bit. The first four of these bits are used to produce the analog sum output, which is done by four *MVke* blocks. The last two bits are used to produce the carry signal, which is done by two more *MVke* blocks.

We note here that the partial products produced by the first step do not all have the same weight. In Figure 9.1 this is graphically represented, as the partialproducts in the bottom four rows have a different alignment as the ones in the top four rows. In order to end up with equal aligned intermediate sums, an adjusted counter can be used for the partial products in the bottom rows.

In counter based binary full-tree multiplication, the number of rows is reduced to two, which subsequently is reduced to one row using a fast adder. However, as opposed to standard adders, EC adders can perform k:1 reduction (within certain limits for k) in almost the same delay as 2:1 reduction [15]. Therefore, in the high-radix EC multiplication scheme the number of rows does not have to be reduced to two, and the reduction process can be stopped earlier than in binary multiplication schemes. For example, in the 8-bit radix 4 EC multiplication in Figure 9.1 only one stage of five counters is required.

The *third step* of the high-radix EC multiplication is the final addition of a number of rows of intermediate sums. In general, this step of the multiplication is performed by some fast addition scheme like carry look-ahead, carry-skip, etc. For the EC paradigm, such a fast addition scheme is not designed yet, thus in this paper we use a ripple carry structured adder.

The addition scheme we use in here, consists of several EC addition blocks, which functions as a high radix, multiple input full adder. The addition block is implemented by an EC counter, but omitting CR^1 and the corresponding $MVke$ blocks. Thus the outputs of the first set of $MPSF$ blocks are producing the output bits. Charge reservoir CR^2 remains and contains the carry signal. To create an adder, the addition blocks are cascaded in parallel with the carry-out of block i connected to the carry-in of block $i+1$.

4.3 Simulation

To verify the high-radix multiplication scheme we simulated the 8-bit radix 4 multiplier. We used an approach of partitioning to simulate the whole multiplier, for the following reason. Although SIMON contains the OpAmp as circuit element, using it in SET circuitry causes some random effects to occur. Partitioning the circuit in parts ending with an OpAmp resolves this problem. To simulate the entire circuit, the output of each OpAmp was stored and used as an input in the next part. The simulation results indicated that the multiplier functions correctly. The multiplier requires 2372 circuit elements and has a total delay of 63.6 *ns*.

5 Conclusions

In this paper we presented a high radix EC addition scheme and a high radix EC multiplication scheme. For both arithmetic operations, first we briefly discussed the normal (non high radix) EC scheme. Second, we presented the high radix addition scheme and explained its functionality. Third, we explained the implementation of the high radix multiplication scheme in details. Finally, we presented simulation results and evaluate the schemes in terms of delay and area cost. The 6-bit radix 8 addition scheme requires 187 circuit elements and has a delay of 13.7 *ns*. The 8-bit radix 4 multiplication scheme requires 2372 circuit elements and has a total delay of 63.6 *ns*.

References

1. Waser, R., ed.: Nanoelectronics and Information Technology - Advanced Electronic Materials and Novel Devices. 1st edn. Wiley-VCH, Berlin (2003)
2. : International Technology Roadmap for Semiconductors, 2003 Edition, Executive Summary. Downloadable from website http://public.itrs.net/Home.htm (2003)
3. Likharev, K.: Single-Electron Devices and Their Applications. Proceeding of the IEEE **87** (1999) 606–632
4. Lageweg, C., Cotofana, S., Vassiliadis, S.: Static buffered set based logic gates. In: 2nd IEEE Conference on Nanotechnology (NANO). (2002) 491–494
5. Cotofana, S., Lageweg, C., Vassiliadis, S.: On computing addition related arithmetic operations via controlled transport of charge. In: proceedings of 16th IEEE Symposium on Computer Arithmetic. (2003) 245–252
6. Cotofana, S., Lageweg, C., Vassiliadis, S.: Addition Related Arithmetic Operations via Controlled Transport of Charge. IEEE Transactions of Computers **54** (2005) 243–256
7. Wasshuber, C.: About Single-Electron Devices and Circuits. PhD thesis, TU Vienna (1998)
8. Ishibashi, K., Tsuya, D., Suzuki, M., Aoyagi, Y.: Fabrication of a Single-Electron Inverter in Multiwall Carbon Nanotubes. Applied Physics Letters **82** (2001) 3307–3309
9. Lageweg, C., Cotofana, S., Vassiliadis, S.: A Linear Threshold Gate Implementation in Single Electron Technology. In: IEEE Computer Society Workshop on VLSI. (2001) 93–98
10. Meenderinck, C.: Single electron tunneling based arithmetic operations. Master's thesis, Delft University of Technology (2005)
11. (http://www.lybrary.com/simon/)
12. Likharev, K., Korotkov, A.: Ultradense Hybrid SET/FET Dynamic RAM: Feasibility of Background Charge Independent Room Temperature Single Electron Digital Circuits. In: Proceedings of the International Semiconductor Device Research Symposium, Charlottesville, Virginia (1995) 355–359
13. Parhami, B.: Computer Arithmetic. Oxford University Press (2000)
14. Dadda, L.: Some schemes for parallel multipliers. Alta Frequenza **34** (1965) 349–356
15. Meenderinck, C., Cotofana, S.D.: Computing periodic symmetric functions in single electron tunneling technology. In: Proceedings of International Semiconductor Conference (CAS). (2005) 47–50