BASIC BUILDING BLOCKS FOR EFFECTIVE SINGLE ELECTRON TUNNELING TECHNOLOGY BASED COMPUTATION

Cor Meenderinck and Sorin Cotofana
Computer Engineering Laboratory
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. Moreover, this new technology opens up avenues for new computational paradigms, which require building blocks with unconventional behavior. In this paper we discuss a number of basic building blocks that allow to effectively implement computational structures in those new paradigms, and analyze them in terms of area, delay, and energy consumption.
Keywords: *single electron tunneling, circuit design, building blocks.*

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 much lower power consumption than CMOS and it is scalable to the nanometer region and beyond [2].

SET technology is fundamentally different from CMOS as it is based on tunneling of electrons. This difference opens up avenues for new computational paradigms of which a number have been proposed [3,4,5], and which try to effectively use the basic SET properties. Theoretical results on the complexity of arithmetic operations using those new paradigms indicate great potential. However, the actual practical results depend on the capabilities of the utilized building blocks. In previous research we already identified a number of such basic building blocks. In this paper we analyze these building blocks with respect to limitations, area, delay, and energy consumption.

This paper is organized as follows. In Section 2 we briefly present some background on SET technology. In Section 3 we present five basic building blocks for SET based computation, and analyze them. Section 4 concludes the paper.

1. BACKGROUND

SET circuits are based on tunnel junctions, which consist of an ultra-thin insulating layer in a conducting material. 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 $V_c = \frac{q_e}{2(C_e + C_j)}$ [6], where q_e=1.602*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 one can describe the switching delay as $t_d = \frac{-\ln(P_{error}) q_e R_t}{|V_j| - V_c}$, 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 Ω 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 SET technology can physically be implemented in various ways, e.g., classical semiconductor lithography and by carbon nanotubes. Therefore, for the blocks we discuss in this paper, the circuit area is evaluated in terms the total number of circuit elements (capacitors and junctions).

2. BASIC BUILDING BLOCKS

SET technology enables accurate control of the transportation of discrete electrons. Moreover, SET allows the representation of values by number of electrons, i.e., Boolean values may be represented by the presence or absence of one electron, while integer values may be represented by the corresponding number of electrons. To effectively utilize this encoding in arithmetic and logic operations, building blocks are required that perform basic signal operations on this Boolean and multi-value signals. Previous investigations suggested that Boolean operations can be implemented using threshold logic gates and inverting buffers [4], while in order to perform arithmetic operations via direct charge

manipulation [3] the following set of building blocks is required: MVke (Move k electrons) block, MCke (Move Conditionally k electrons) block, and PSF (periodic symmetric function) block. This section presents the implementation and analysis of those basic building blocks. The results presented in here are based on calculations and SIMON [7] simulations.

2.1. THRESHOLD GATE

An *n*-input linear threshold logic gate is a device which is able to compute any linearly separable Boolean function given by $F(X)=\mathrm{sgn}\{f(X)\}$ and $f(X)=\sum_{i=1}^{n} w_i x_i - \varphi$, where x_i are the *n* Boolean inputs and w_i are the corresponding *n* integer weights. The linear threshold gate performs a comparison between the weighted sum of the inputs $\sum_{i=1}^{n} w_i x_i$ and the threshold value φ. If the weighted sum of its inputs is *greater than or equal to* the threshold, the gate produces a logic '1'. Otherwise the output is a logic '0'. The threshold logic gate can operate on Boolean signals as well as on multi valued digital or analog signals. Figure 1 depicts an implementation of the threshold gate in SET technology.

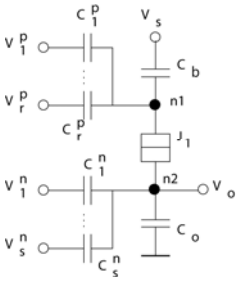

Fig. 1. Threshold logic gate.

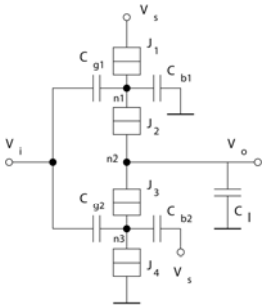

Fig. 2. Inverting buffer.

Using the circuit presented above, theoretically we can create threshold gates with an arbitrary number of inputs. However, for practical circuits the size of the capacitors C_i^p and C_i^n as well as the desired output voltage restrict the possibilities. The exact limitations are dependent on the actual implementation.

As the energy consumption and the delay are dependent on the voltage across the junction at the time of a tunnel event, they are thus implementation dependent too. For a typical implementation of a 2-input Boolean gate the energy consumption is approximately 0.8 meV and the delay is approximately 0.8 ns. In this paper a typical implementation assumes a supply voltage of 16 mV and a representation of logic '1' of 16 mV as well. To get a better inside in this matter we assessed the dependence between delay, energy, and the number of inputs, of which the results are presented in Figure 3. The energy consumption is linear with the number of inputs. However, the delay is independent on the number of inputs, in contrast to CMOS logic gates where we see such a dependency.

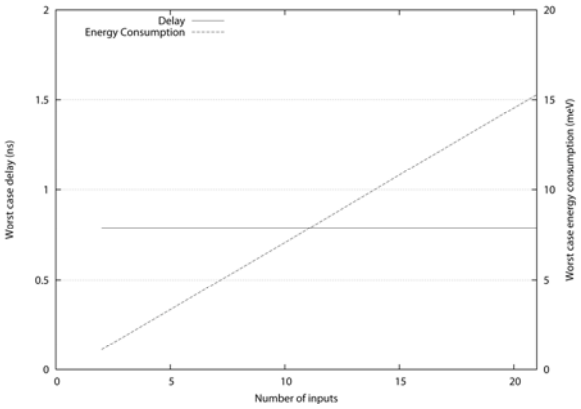

Fig 3. Delay and energy consumption of an n-input AND gate implemented by a SET threshold gate.

2.2. THE INVERTING BUFFER

To improve the fan-out capabilities and to reduce feedback effects, the threshold logic gate has to be augmented with a buffer. Figure 2 depicts a possible implementation of such an inverting buffer. Other implementations are possible, but this one is preferred as it has the lowest output to input feedback ratio, which is fundamental for a buffer. The circuit consists of two SET transistors which are augmented with a bias capacitance (C_{b1} and C_{b2}) and an output capacitor C_l. The upper SET transistor (J1 and J2) operates similar to a p-type CMOS transistor while the lower SET transistor (J3 and J4) operates like an n-type CMOS transistor.

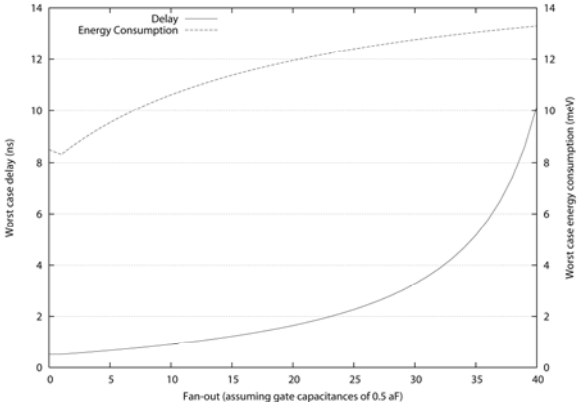

Fig. 4. Delay and energy consumption of the inverting buffer with respect to fan-out.

The area cost of the inverting buffer is 9 elements. When designing such a buffer, there are a number of trade offs that can be made with respect to delay, energy consumption, feedback ratio, and sensitivity to imperfections. A number of different designs have been implemented and simulated. The delay ranges from 0.4 ns to 1.9 ns while the energy consumption ranges from 7 meV to 11 meV, assuming a fan-out of

one. Figure 4 depicts the dependency of the delay and energy consumption on the fan-out of the buffer. For this experiment we assumed that all gates on the output of the buffer have an input capacitance equal to that of the buffer itself, i.e., 0,5 aF. From the graph we can see that the delay of the buffer doubles for every 11 extra gates on the output. The maximum number of gates that the buffer can drive is depending on the amount of feedback that the gates cause on the output node of the buffer. Our experiments indicate that for a typical implementation of the buffer, it can drive up to 19 threshold logic gates.

2.3. MVke

The Move *k* electron (*MVke*) block controls the transport of an adjustable number of electrons to/from a charge reservoir. An *MVke* block has inputs V_e (enable), V_r (reset), and V_v (V) and has a build in constant *k* such that the circuit transports V*k* electrons when enabled. When a reset is applied all electrons return to their original position and the circuit becomes charge neutral.

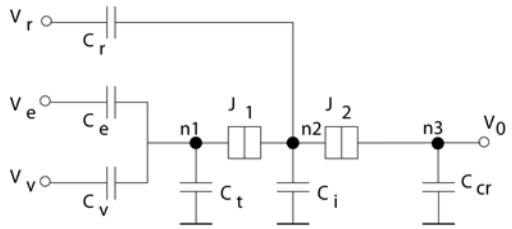

Fig. 5. MVke block

Figure 5 presents an *MVke* implementation that can remove electrons from a charge reservoir, which is implemented by a large capacitance (C_{cr}).

The number of transportable electrons is limited by the voltage they produce on the output reservoir. Implementing the charge reservoir with a large capacitance, results in a low voltage on the output of the *MVke* block and thus in a high limit to the number of transportable electrons. On the other hand, the low voltage on the charge reservoir requires a higher accuracy in the next stage, which operates on the value in the reservoir. For a typical implementation of the *MVke* block we calculated an upper limit to the number of transportable electrons of 529.

The area cost of the *MVke* block is 8 elements. The delay and the energy consumption are dependent on the actual parameters of the circuit and on the actual number of transported electrons. Figure 6 depicts the delay and energy consumption for different values of transported electrons *k*, assuming a typical implementation. As expected, the energy consumption is linear to the number of transported electrons. The delay is logarithmic to the number of transported electrons, because for large *k* the first electrons experience a larger 'force' and therefore tunnel a lot faster.

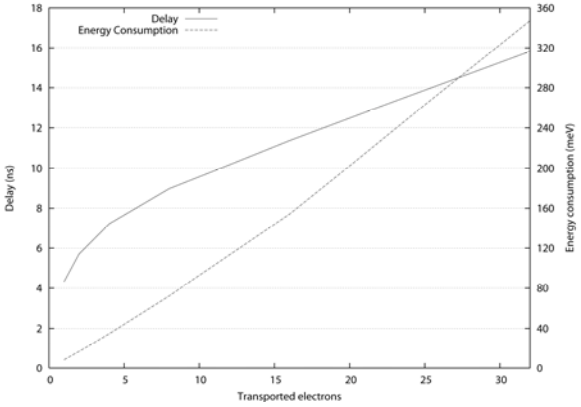

Fig. 6. Delay and energy consumption of the MVke block.

2.4. MCke

The Move Conditional *k* electrons (*MCke)* block transports a fixed amount of electrons *k* to or from a charge reservoir if and only if the input V_v exceeds a certain threshold. The *MCke* block also has an enable (V_e) input and a reset (V_r), thus it is capable of returning the transported electrons.

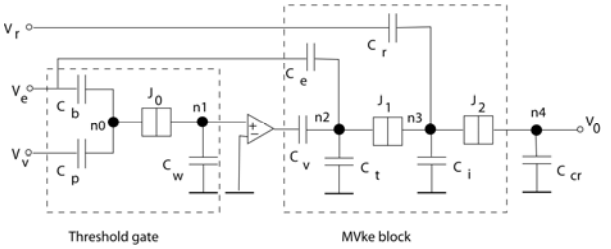

Fig. 7. The MCke block.

If we split the functionality of the *MCke* block in two parts, i.e., detecting the threshold condition and moving electrons, it is clear that the *MCke* block can be implemented by using a SET threshold gate in combination with an *MVke* block. The implementation depicted in Figure 7 uses this approach. The output of the threshold gate is buffered by an OpAmp, which is required to guaranty correct functionality of both building blocks. We note here that OpAmps can potentially be implemented using a hybrid FET-SET technology [8].

The area cost of the *MCke* block is 12 elements. Assuming a typical implementation and a value for *k* of 1, the delay is 1.55 ns and the energy consumption is 10.3 meV. These values are excluding the area, delay and energy consumption of the required OpAmp. The delay of the threshold logic gate is very much depending on the voltage difference it has to detect. For large steps on the input voltage the delay is very small. However, for small steps on the input the delay can be very large. Figure 10 depicts the delay and energy consumption of the threshold logic gate, the

MVke block, and their total. For this experiment we assumed, for the *MVke* block a value for *k* of 1.

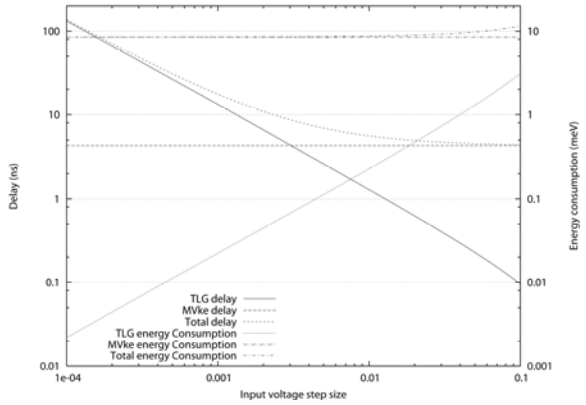

Fig. 8. Delay and energy consumption of the MCke block with respect ot the input step size.

2.5. PSF

A *PSF* building block implements a Periodic Symmetric Function on one or multiple inputs. A Boolean function of *n* variables F_s, is symmetric if and only if for any permutation σ of < 1,2,...,*n* >, $F_s(x_1, x_2, ..., x_n) = F_s(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$. In other words, a Boolean symmetric function entirely depends on the sum of its input values $F_s(x_1, x_2, ..., x_n) = F_s\left(\sum_{i=1}^{n} x_i\right)$. A periodic symmetric function is a symmetric function for which there exists a period *T* such that $F_s(X) = F_s(X + T)$.

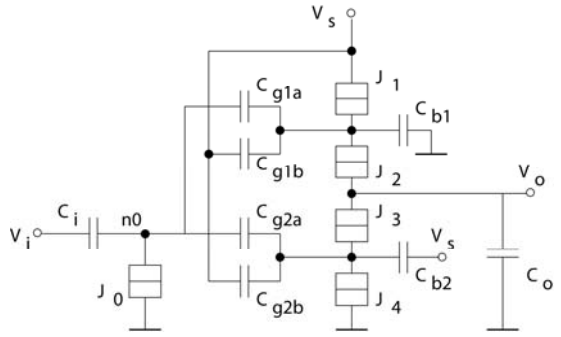

Fig.9. The PSF block.

A *PSF* block can be implemented using a SET electron trap (C_i and *J0*) in combination with a SET inverter (see Figure 9). The electron trap has a triangular periodic transfer function. The inverter acts in this case as a literal gate and transforms the triangular transfer function into a rectangular shape.

The *PSF* block has an area cost 12 elements plus one for each input. The delay and energy consumption of the *PSF* block are dependent on the number and the weight of the inputs. For typical implementations the delay ranges from 1.5 ns to 10 ns while the energy consumption ranges from 8 to 90 meV. Figure 10 presents the delay and energy consumption of a *PSF* block with respect to the input value. The *PSF* block implements a periodic symmetric function with a period of 2, which is also reflected in the delay. The energy consumption is exponential to the input value.

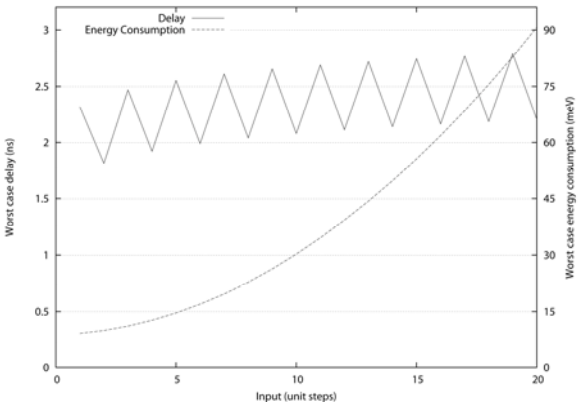

Fig. 10. Delay and energy consumption of the PSF block.

4. CONCLUSIONS

In this paper we presented a set of building blocks, based on SET technology devices, that allows the efficient implementation of computational structures in new paradigms. The proposed building blocks can operate on Boolean signals (one electron encoded signals) as well as on discreet analog signals (multiple electron encoded signals) and can perform conversions among them. We analyzed the building blocks in terms of area, delay, and energy consumption and discussed a number of design trade offs. This study provides us the means to evaluate the actual expected performance of SET based schemes for a given fabrication technology and error probability.

REFERENCES

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

[2] "International Technology Roadmap for Semiconductors, 2003 Edition, Executive Summary".

[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. Lageweg, "Single electron tunneling based arithmetic computation," Ph.D. dissertation, TU Delft, 2004.

[5] N. Asahi, M. Akazawa, and Y. Amemiya, "Single-Electron Logic Systems Based on the Binary Decision Diagram," IEICE Transactions on Electronics, vol. E81-C, no. 1, pp. 49–56, January 1998.

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

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

[8] K. Likharev and A. Korotkov, "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, October 1995, pp. 355–359.