Single Electron Encoded SET Memory Elements

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

Abstract— **The ability to control the transport of individual electrons in Single Electron Tunneling (SET) based circuits creates the conditions for Single Electron Encoded Logic (SEEL). The logic values 0 and 1 of SEEL gates are encoded as a net charge of 0 or 1 electron charges. This paper investigates the implementation of SEEL memory elements based on SEEL Boolean logic gates. After the introduction of the SEEL Boolean logic gates that serve as basic building blocks, SEEL implementations of the RS-latch, D-latch and positive edge-triggered D flip-flop are proposed and verified by simulation. Finally, the area, switching delay and power consumption of the memory elements are compared with CMOS-like SET transistor based implementations. The comparison indicates that the total circuit area is comparable, but the SEEL Boolean gate based implementations reduce both delay and power consumption by a factor of 5.**

Keywords— **single electron technology, SET, single electron encoded logic, memory elements**

I. Introduction

In recent years the notion has reemerged that feature sizes in semiconductor-based integrated circuits cannot be shrunk indefinitely (see for example [12]). Given this observation, it is of interest to investigate possible alternative technologies for the future [8]. Devices based on Single Electron Tunneling (SET) [7], [6] promise excellent potential for future Ultra Large Scale Integrated (ULSI) circuits due to their potential for low power consumption and feature size scalability. Additionally, assuming that circuit features in the order of nanometers or less can be produced, the SET technology can be applied at room temperature. This is rapidly becoming a possibility due to recent advances in silicon based fabrication technology (see for example [11]).

Most studies carried out on SET based logic circuits assume that the tunnel junction operates as a switch, and use it to implement the SET equivalent of the MOS transistor [1], [5]. Although this has the advantage that existing CMOS transistor-based designs can easily be ported to SET technology, it does not fully utilize the potential of the SET technology. The main disadvantage of the CMOS like design style is that the current transport though an "open" transistor still consists of a large number of individual electrons "dripping" through the tunnel junctions. However, SET circuits are centered around the tunnel junction, through which individual electrons can be transported in a controlled manner. This creates the conditions for encoding the Boolean logic values 0 and 1 as a charge of 0 or 1 electron charges stored on a circuit's output node, thus for realizing Single Electron Encoded Logic (SEEL) circuits. SEEL circuits can be realized if, starting from a neutral charge distribution, a single electron charge can be removed from or added to a circuit's output node. Gates that operate on the SEEL paradigm are potentially faster while also consuming less power, since only 1 electron transport is required for output switching.

Earlier investigations revealed that static buffered SEEL logic gates can be constructed based on the Coulomb blockade of SET tunnel junctions [4]. A family of static buffered SEEL Boolean logic gates was proposed [3], which was demonstrated to operate correctly for logic and arithmetic circuits. However, practical designs have to be able to exhibit sequential behavior and thus require both logic gates and memory elements, e.g., latches and flip-flops. In this paper we investigate the implementation of SEEL memory elements using static buffered SEEL Boolean logic gates as basic building blocks. First, we propose implementations for the R-S latch, the D latch, and the edge triggered D flip-flop. Second, we verify that the proposed memory elements perform the correct logic function by means of simulation. Third, we compare the area, switching delay and power consumption of the proposed memory elements with CMOS-like SET based implementations.

The remainder of this paper is organized as follows. Section II briefly presents the SET background theory. Section III introduces the static buffered SEEL Boolean logic gates that are utilized as building blocks for the SEEL memory elements. The proposed R-S latch, D latch, and edge triggered D flip-flop implementations, as well as simulation results, are presented in Section IV. Section V compares the area, switching delay and power consumption of the proposed memory elements with CMOS-like SET implementations. Finally, section VI concludes the paper.

II. Background of Single Electron Tunneling

A tunnel junction can be thought of as a leaky capacitor. The transport of charge through a tunnel junction is referred to as *tunneling*, where the transport of a single electron through a tunnel junction is referred to as a *tunnel event*. Electrons are considered to tunnel through a tunnel junction strictly one after another. The critical voltage V_c across a tunnel junction is the voltage threshold needed across the tunnel junction in order 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 [9], we calculate the critical voltage 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 $q_e = 1.602 * 10^{-19}$ C. Strictly speaking this is incorrect, as the charge of the electron is of course negative. However, it is more intuitive to consider e as a positive constant for the formulas which determine whether or not a tunnel event will occur. 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 will occur through this tunnel junction if and only if:

$$|V_j| \geq V_c \tag{2}$$

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 only consider 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.

The transport of an electron through a tunnel junction is a stochastic process. This means that we cannot analyze delay in the traditional sense. Instead, assuming a non-zero probability for charge transport ($|V_j| > V_c$), the switching delay t_d of a single electron transport can be calculated based on an error probability P_{error} that the desired transport did *not* occur as

$$t_d = \frac{-ln(P_{error})q_e R_t}{|V_j| - V_c} \tag{3}$$

where $R_t = 10^5\Omega$ is the tunnel resistance (though depending on the physical implementation this value is typically assumed). The error probability P_{error} will determine the reliability of the circuit. Given that the switching behavior is stochastic in nature, the error probability cannot be reduces to 0. It is therefore assumed that an error correction mechanism will be present in the form of hardware or data redundancy in order to achieve the desired accuracy. In the remainder of this paper we assume that $P_{error} = 10^{-12}$, which is sufficiently small for such a mechanism to be applied successfully.

When charge transport occurs through a tunnel junction, the difference in the total amount of energy present in the circuit before and after the tunnel event can be calculated by

$$\Delta E = E_{final} - E_{initial} = -q_e(|V_j| - V_c) \tag{4}$$

Therefore, the energy consumed by a single tunnel event occurring in a single tunnel junction can be calculated by taking the absolute value of ΔE. In order to calculate the power consumption of a gate, the energy consumption of each tunnel event is multiplied by the frequency of switching. The switching frequency in turn depends on the frequency at which the gate's inputs change and is input data dependent, as a new combination of inputs may or may not results in charge transport.

Given that the implementations proposed in this work are intended to be general, i.e., independent of a manufacturing process, we only estimate the area of circuits in terms of the number of required circuits element. By circuit elements, we refer to the number of capacitors and tunnel junctions required for each implementations.

III. Building Blocks

Threshold logic gates are devices which are able to compute any linearly separable Boolean function given by:

$$\begin{aligned} Y &= sgn\{\mathcal{F}(X)\} = \begin{cases} 0 & \text{if } \mathcal{F}(X) < 0 \\ 1 & \text{if } \mathcal{F}(X) \geq 0 \end{cases} & (5) \\ \mathcal{F}(X) &= \sum_{i=1}^{n} \omega_i x_i - \psi & (6) \end{aligned}$$

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 $\Sigma_{i=1}^{n} \omega_i x_i$ and the threshold value ψ. If the weighted sum of inputs is *greater than or equal to* the threshold, the gate produces a logic 1. Otherwise the output is a logic 0. A generic threshold gate scheme [2], which is displayed by Figure1, has been proposed. This scheme can be used as a basis for implementing linear threshold gates with both positive and negative weights.

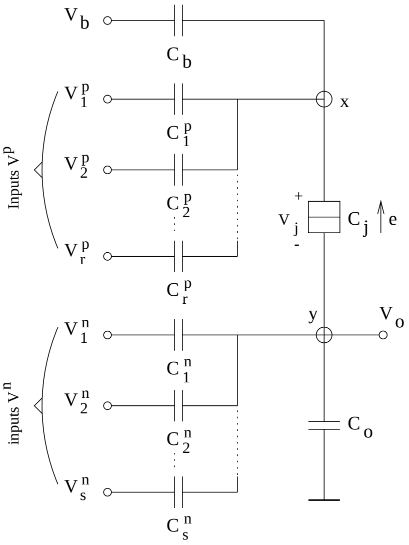

Fig. 1. SET generic linear threshold gate.

Such threshold gates however do not operate correctly in networks due to the passive nature of the circuit. It was found [4] that augmenting the output of each threshold gate with a SET buffer/inverter consisting of 2 SET transistors, as displayed by Figure 2(b), results in correctly operating threshold gate networks. The buffer can also function as a stand-alone inverter gate. Both the generic threshold gate and the buffer operate in accordance with the single electron encoded logic (SEEL) paradigm, i.e., charge transport due to switching activity is limited to 1 electron.

Given that the basic Boolean logic functions AND, OR, NAND, and NOR can be specified in the form of Equations (5,6), we can implement the AND, OR, NAND, and NOR gates as instances of the generic threshold gate circuit (displayed in Figure 1) augmented with a buffer/inverter (displayed in Figure 2. In theory, the thresholds ψ are integer numbers. However, if the threshold for example is $\psi = i$ (i being an integer value), this implies that the gates function correctly for any value in the interval $i - 1 < \psi <= i$. In order to maximize robustness for variations in parameter values, we replace the threshold value $\psi = i$ by the average $\psi = i - \frac{1}{2}$. The correctness of the above can easily be verified. Consequently, the threshold equations of the 2-input AND, OR, NAND and NOR gates can be written as:

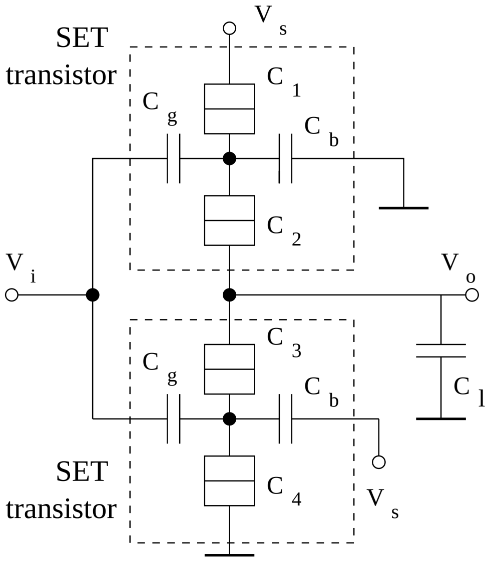

Fig. 2. SET buffer/inverter circuit implementation.

$$\begin{aligned} AND(a,b) &= sgn\{a+b-1.5\} && (7)\\ OR(a,b) &= sgn\{a+b-0.5\} && (8)\\ NAND(a,b) &= sgn\{-a-b+1.5\} && (9)\\ NOR(a,b) &= sgn\{-a-b+0.5\} && (10) \end{aligned}$$

The threshold gate based implementations of the Boolean gates all have the same basic circuit topology (see Figure 1 for general case), consisting of a bias capacitor C_b, a tunnel junction with capacitance C_j and an output capacitor C_o. The AND and OR gates contain two input capacitors $C_1^p = C_2^p = 0.5C$ for positively weighted inputs, while the NAND and NOR gates contain two capacitors $C_1^n = C_2^n = 0.5C$ for negatively weighted inputs. Additionally, each of the threshold gates is augmented with an output buffer. Given that the buffer inverts its input, the logic function performed by the buffered threshold gate is the inverse of that performed by the threshold gate itself. For example, a buffered NAND gate consists of a threshold gate that implements the AND function and a buffer. For the remainder of this discussing, when referring to a logic gate such as a NAND gate, we imply the logic function performed by the entire gate (threshold gate + output buffer).

In the remainder of this paper we assume that Boolean input/output signals correspond with the following voltages: logic 0 = 0 Volt, logic 1 = $0.1q_e/C$ Volt, where C acts as a unit for capacitance. For the circuit simulations it is assumed that $C = 1aF$, resulting in logic 1 = $16mV$. For the buffer/inverter part of the logic

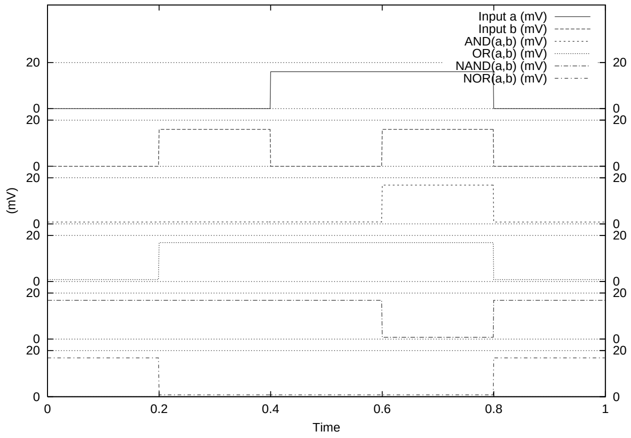

Fig. 3. Simulation results for 2-input SEEL Boolean gates.

gates the following circuit parameter ratios are assumed: $C_g = 0.5C$, $C_b = 4.25C$, $C_1 = C_4 = 0.1C$, $C_2 = C_3 = 0.5C$, $C_l = 9C$, $V_s = 0.1q_e/C$. For the threshold gate part of the logic gates the following circuit parameter ratios are assumed: $C_b(AND)$ = $10.6C$, $C_b(OR)$ = $11.7C$, $C_b(NAND)$ = $13.2C$, $C_b(NOR)$ = $11.7C$, C_j = $0.1C$, $C_1^p(NAND, NOR)$ = $C_2^p(NAND, NOR)$ = $0.5C$, $C_1^n(AND, OR)$ = $C_2^n(AND, OR)$ = $0.5C$, $C_o(AND, OR)$ = $8C$, $C_o(NAND, NOR)$ = $9C$. The resulting SEEL Boolean logic gates have been verified by means of simulation using the single-electron device and circuit simulator SIMON (Simulation Of Nanostructures) [10]. The simulation results are depicted in Figure 3. As can be observed, each of the gates correctly implements the specified logic function.

Gate	Area	Delay	Energy
2-input AND	14 elements	1.7 ns	10.8 meV
2-input OR	14 elements	1.7 ns	10.8 meV
2-input NAND	14 elements	2.2 ns	10.7 meV
2-input NOR	14 elements	2.0 ns	10.7 meV
Inverter	9 elements	0.6 ns	10.4 meV

TABLE I
AREA, DELAY AND ENERGY CONSUMPTION.

Given the circuit parameters ratios for the AND, OR, NAND and NOR gate described above, as well as those for the stand-alone inverter, and assuming $C = 1aF$, we calculated the area, delay and power consumption of each of the gates. The combined results are summarized in Table I. We emphasize that these gates were neither optimized for delay nor for power consumption. Also, given the capacitor ratio's and voltage levels which were used for the 2-input buffered Boolean gates, reducing the unit for capacitance C by 1 order of magnitude has the effect of reducing delay by 1 order of magnitude while increasing power consumption by 1 order of magnitude.

IV. SINGLE ELECTRON MEMORY ELEMENTS

In this section we investigate Boolean gate based implementations of the R-S latch, the D latch and the D flip-flop. Each of these implementations is based on the family of SEEL Boolean logic gates described in the previous section.

A. R-S Latch Implementation

The R-S (reset-set) latch is a memory element with 2 inputs (R and S) and 2 outputs (Q and QN). The behavior of the R-S latch is as follows. If R and S are both 0, the R-S latch holds the current output values. If $S = 1$ and $R = 0$, the outputs are set to $Q = 1$ and $QN = 0$. If $R = 1$ and $S = 0$, the outputs are reset to $Q = 0$ and $QN = 1$. The remaining input combination $R = S = 1$ should be avoided during normal operation, as for Boolean gates based implementations it typically results in unstable output values.

A Boolean gate based implementation of the R-S latch usually consists of 2 cross-coupled gates that form a feedback loop. An R-S latch implementation based on NOR gates is depicted in Figure 4. The circuit operates as follows. When R and S are both 0, the 2 gates behave as chained inverters and form a bi-stable element (where $Q = 0, QN = 1$ and $Q = 1, QN = 0$ are stable states). If $R = 1$ while $S = 0$, the output of the upper NOR gate is forced to 0, resulting in $Q = 0$ and $QN = 1$ (similar for $S = 1$ and $R = 0$, resulting in $Q = 1$ and $QN = 0$). If $R = S = 1$, the output of both NOR gates is forced to 0. If both inputs are then dropped to 0 simultaneously (forming a bi-stable element), the circuit either switches to $Q = 0, QN = 1$ or $Q = 1, QN = 0$, and may even oscillate between these two solutions. Given that this behavior is unpredictable, this input combination should be avoided.

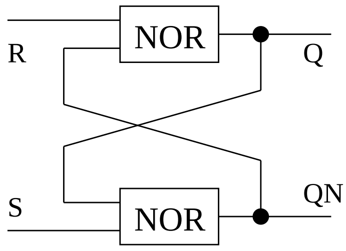

Fig. 4. Boolean gate based R-S latch implementation.

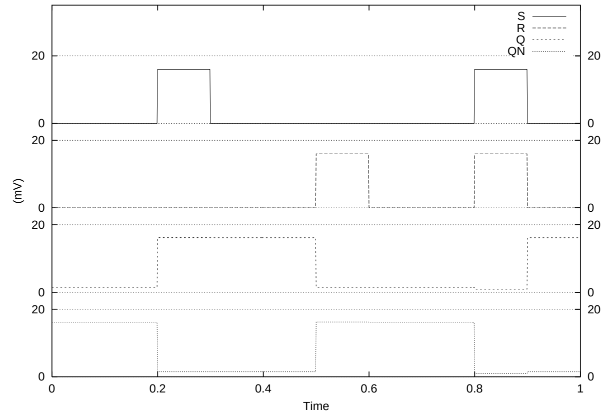

Fig. 5. Simulation results of Boolean gate based R-S latch.

A NOR gate based R-S latch implementation has been verified by means of simulation (using the simulation package SIMON [10]). The circuit parameters of the 2 NOR gates are as specified in Section III. The simulation results are depicted in Figure 5. The first two bars represent the inputs R and S, the bottom two bars the outputs Q and QN. Initially, the inputs are $S = 0$ and $R = 0$, while the outputs are $Q = 0$ and $QN = 1$. When the input S becomes 1, the outputs are set to $Q = 1$ and $QN = 0$. These output values are memorized when S returns to 0. Likewise, when R becomes 1, the outputs are reset to $Q = 0$ and $QN = 1$, which is memorized when R return to 0. Next, R and S are both set to 1, as a result of which the outputs Q and QN are both set to 0. Note that the outputs Q and QN should normally be their logic counterparts in order to prevent possible unstable behavior. When R and S are then simultaneously set to 0, the simulator evaluates possible tunnel events until all are resolved, which for this simulation resulted in the displayed output values $Q = 1$ and $QN = 0$. We therefore conclude that the circuit correctly implements the behavior of an R-S latch as described above.

B. D Latch Implementation

The D latch is a memory element with 2 inputs (D and L) and 2 outputs (Q and its complement QN). The behavior of the D latch is as follows. If the input $L = 0$, the D latch holds the current output values. If the input $L = 1$, the latch is transparent and the output Q follows the input D (and QN follows the complement of D). Unlike the R-S latch, the D latch does not have an unspecified or forbidden input combination.

A possible D latch implementation based on Boolean logic gates is depicted in Figure 6. The circuit operates as follows. The cross-coupled NAND gates form an $\overline{R}$-$\overline{S}$ latch, where $\overline{R} = \overline{S} = 1$ corresponds with the hold function. When $L = 0$, the inputs of the $\overline{R}$-$\overline{S}$ are both 1 regardless of the value of D, and the $\overline{R}$-$\overline{S}$ latch holds its current output values. If $L = 1$, the inputs of the $\overline{R}$-$\overline{S}$ have complementary values, and the output Q becomes follows the value of D.

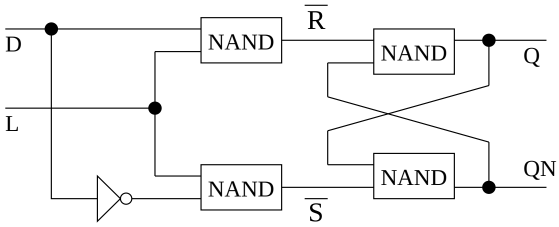

Fig. 6. Boolean gate based D-Latch implementation.

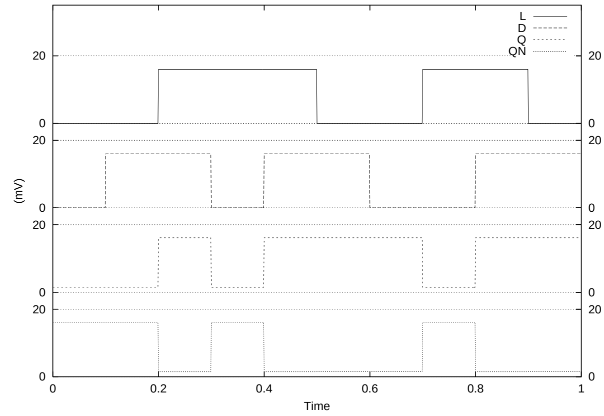

Fig. 7. Simulation results of Boolean gate based D latch.

A D latch implementation consisting of the Boolean logic gates discussed in Section III has been verified by means of simulation (using SIMON). Note that the inverter gate is a stand-alone buffer. The simulation results are depicted in Figure 5. The first two bars represent the inputs D and L, the bottom two bars the outputs Q and $\overline{Q}$. Initially, the inputs are $L = 0$ and $D = 0$, while the outputs are $Q = 0$ and $QN = 1$. When D becomes 1 while L remains 0, the outputs remain unchanged. Once L is set to 1, the output Q follows the values of D until L is dropped to 0. At that point the last values of Q is memorized. The same can be observed when L becomes 1 again. We therefore conclude that the circuit correctly implements the behavior of the D latch as described above.

C. D Flip-Flop Implementation

A D flip-flop is a memory element with 2 inputs (D and CLK) and 2 outputs (Q and its complement QN). The behavior of a negative edge-triggered D flip-flop is as follows. When the input CLK transitions from 1 to 0 (a falling or negative edge on the time graph), the flip-flop samples the current value of D and copies this value to the output Q. For all other input combinations, including a positive-edge clock transition from 0 to 1, the circuit holds its current output values.

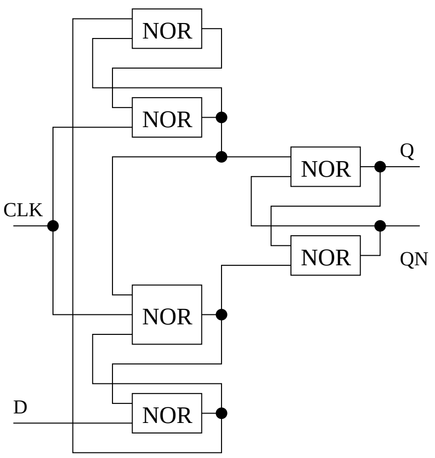

Fig. 8. Boolean gate based negative edge-triggered D flip-flop.

A straightforward implementation of an edge-triggered D flip-flop consists of a chain of 2 D latches. Given that this D latch implementation requires 4 NAND gates and 1 inverter, such an implementation would require at least 8 NAND gates and 2 inverters. However, there are faster and smaller Boolean gate based implementations that specifically make use of the R-S latch' 'unstable' output values (such as $Q = 0$ and $QN = 0$ for the cross-coupled NOR gate based R-S latch). One such implementation, realizing a negative edge-triggered D flip-flop, is based on 3 NOR gate based R-S latches. This flip-flop implementation requires a total of 5 2-input NOR gates and 1 3-input NOR gate as depicted in Figure 8.

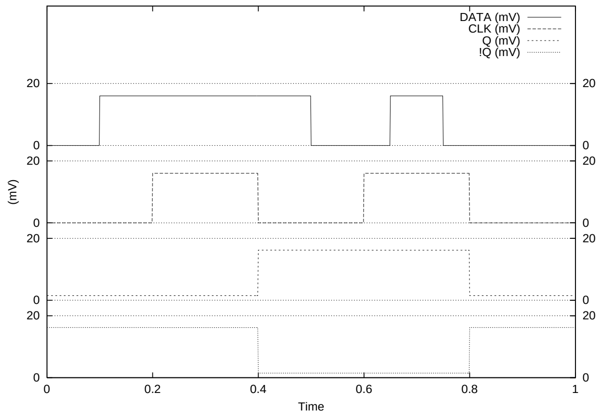

Fig. 9. Simulation result of negative edge-triggered D flip-flop.

The negative edge-triggered D flip-flop has been implemented using the buffered Boolean logic gates discussed in Section III. The 3-input NOR gate was also derived as an instance of the generic threshold gate (as a 3-input OR gate) and then augmented with a static inverting buffer. Given the same methodology as described for the 2-input gates in Section III, we calculated $C_b(3 - input\ NOR) = 12.3C$ while all other circuit parameters remain as used for the 2-input NOR.

We have simulated the negative edge-triggered D flip-flop circuit using the simulation package SIMON. The simulation results are displayed in Figure 9. Starting from the top of the figure, the first row represent the input data signal D. The second row represents the clock signal CLK. The third and fourth bar represent the two outputs Q and QN of the flipflop. Initially, the inputs are $D = 0$ and $CLK = 0$, while the outputs are $Q = 0$ and $QN = 1$. When the input D changes to 1, the output remain unchanged. The same applied when the input CLK changes to 1. The input D is sampled for the first time when CLK changes back to 0 (a negative-edge), resulting in $Q = 1$ and $QN = 0$. These output values are memorized until D is sampled again during the next negative-edge transition of CLK, at which point the outputs become $Q = 0$ and $QN = 1$. We therefore conclude that the negative edge-triggered D flip-flop operates correctly, sampling the input D (and updating outputs Q and QN) only on the negative edge of the clock signal CLK.

V. Area, Delay and Energy Consumption

In this section we estimate the area, delay and energy consumption of the memory elements proposed in Section IV. We compare our results with a CMOS-like SET transistor based approach, as originally proposed by [5].

We can estimate the area, delay and power consumption of the SEEL Boolean gate based memory element implementations discussed in Section IV by adding the results obtained for the individual gates. The area, delay and power consumption of the inverter gate and the 2-input buffered Boolean logic gates are presented in Table I. Given this approach, the obtained results for the Boolean gate based implementations are presented in Table II.

The CMOS-like SET transistor based approach results in Boolean logic gates whose gate layout is similar to the basic CMOS cell layout. For the 2-input AND, OR, NAND and NOR gates, this results in a 4 transistor design (plus some additional load capacitors). A single SET transistor requires 4 circuit elements. Thus, these gates require a minimum total area of 16 circuit elements. Given that we have not implemented the Boolean gates based on CMOS-like SET transistor, we utilize this area estimate as a lower bound.

The switching delay and energy consumption of the

Circuit	Area	Delay	Power
R-S latch	28	4.0 ns	21.4 meV
D latch	65	7.2 ns	53.2 meV
D flip-flop	85	6.0 ns	64.2 meV

TABLE II
AREA, DELAY AND ENERGY CONSUMPTION OF SEEL BOOLEAN GATE BASED IMPLEMENTATIONS.

Circuit	Area	Delay	Power
R-S latch	32	23.2 ns	112.4 meV
D latch	73	35.4 ns	235.2 meV
D flip-flop	96	34.8 ns	337.2 meV

TABLE III
AREA, DELAY AND ENERGY CONSUMPTION OF CMOS-LIKE SET TRANSISTOR BASED IMPLEMENTATIONS.

CMOS-like SET transistor based gates are estimated based on charge transport through a single SET transistor. Unlike the SEEL gates, the SET transistor based designs require the transport of a large number of electrons (typically at least 100) in order to realized output switching. However, in order to compare the CMOS-like approach to the SEEL approach, we assume that the circuit parameters of the individual SET transistors are equal to those used for the SEEL inverter depicted in Figure 2. The SEEL buffer is based on a much smaller load capacitor, which implies that the transport of a smaller amount of electron should be sufficient to accomplish output switching. We therefore assume that charge transport through the SET transistors of CMOS-like SET transistor based gates gate consists of only 10 electrons. In CMOS-like designs it is often sufficient that a gate that switches by 80% of its maximum output swing. We therefore can assume that the transport of 8 of these electrons is sufficient in order to realized output switching. Also, in order to simplify the derivations we assume that the charge transport occurs through only 1 transistor (and not through 2 transistors in series or in parallel). Finally, we assume that the voltage levels and circuit parameters of the transistor through which the charge transport occurs correspond with those of the buffer/inverter depicted in Figure 2.

Consequently, the delay and consumed energy of the first electron transported through the SET transistor is equal to that of the buffer/inverter as stated in Table I. Consecutive charge transports will result in a linear reduction of voltage across the SET transistor's tunnel junctions, until it is less then the critical voltage required for tunneling. In other words, during the transport of the 10 electrons $|V_j| - V_c$ will linearly reduce from its initial value to approximately 0. Resulting, we calculate that transporting 8 electrons requires 19.3 times as much delay and 5.4 times as much energy as transporting the first electron. We therefore estimate for each of the CMOS-like SET transistor based 2-input Boolean logic gates that $t_d = 11.6ns$ and $\Delta E = 56.2meV$. Given these estimates, the implementation of the R-S latch, D latch and D flip-flop based on CMOS-like SET transistor based Boolean logic gates are presented in Table III.

Comparing the results of the two design styles, we observe the following. The required area is comparable, although a lower bound estimate has been used for the CMOS-like SET transistor based approach. Thus in practice the SEEL Boolean gate based designs require less area. On the other hand, both the delay and power consumption of the SEEL Boolean gate based approach are about 5 times less than that required for the CMOS-like SET transistor based approach.

VI. CONCLUSIONS

This paper investigated the implementation of Single Electron Encoded Logic (SEEL) memory elements based on SEEL Boolean logic gates. After introduction of the SEEL Boolean logic gates that serve as basic building blocks, SEEL implementations of the RS-latch, D-latch and positive edge-triggered D flip-flop were proposed and verified by simulation. Finally, the area, switching delay and power consumption of the memory elements were compared with CMOS-like CMOS-like SET transistor based implementations. We found that the total circuit area is comparable, but the SEEL Boolean gate based implementations reduce both delay and power consumption by a factor of 5.

REFERENCES

[1] A.Korotkov, R.Chen, and K.Likharev. Possible Performance of Capacitively Coupled Single-Electron Transistors in Digital Circuits. *Journal of Applied Physics*, 78(4):2520–2530, August 1995.

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

[3] C. Lageweg and S. Cotofana and S. Vassiliadis. A family of single electron static buffered Boolean logic. In *Proceedings of ProRISC 2002*, pages 339–343, November 2002.

[4] C. Lageweg and S. Cotofana and S. Vassiliadis. Static Buffered SET Based Logic Gates. In *2nd IEEE Conference on Nanotechnology (NANO)*, pages 491–494, August 2002.

[5] J.R.Tucker. Complementary Digital Logic based on the ”Coulomb Blockade”. *Journal of Applied Physics*, 72(9):4399–4413, November 1992.

[6] A. Korotkov. Single-Electron Logic and Memory Devices. *International Journal of Electronics*, 86(5):511–547, 1999.

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

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.