Proceedings of the 7th IEEE
International Conference on Nanotechnology
August 2 - 5, 2007, Hong Kong

Building Blocks for Delay-Insensitive Circuits using Single Electron Tunneling Devices

Saleh Safiruddin and Sorin D. Cotofana, *Senior Member, IEEE*

Abstract — **This paper presents a set of basic building blocks that corresponds to a universal set of primitives for delay insensitive circuits. We propose single electron tunneling circuit topologies and verify them by means of simulations. The simulations performed with SIMON 2.0 indicate that the circuits function as expected. Moreover the proposed circuits are input-output level compatible thus they can be potentially utilized in the implementation of larger asynchronous circuits.**

I. BACKGROUND

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 to CMOS 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 [3, 4, 5, 6], 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, electron tunneling is stochastic in nature. Tunneling through a junction becomes possible when the junction's current voltage V_j exceeds the junction's critical voltage $V_c = \frac{q_e}{2(C_e + C_j)}$ [7], 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. The delay of such circuits 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. This probabilistic delay makes difficult the direct utilization of SET based computation in building synchronous computation units.

An alternative solution for SET based computation is to utilize Delay-Insensitive Circuits (DIC), which by their very nature do not require a synchronization signal thus can naturally tolerate the probabilistic behavior of SET circuits. Previous work [9] has already determined a universal set of basic primitives from which, theoretically speaking, any asynchronous circuit can be built. Those primitives are described only at behavioral level and no implementations have been proposed yet. While building blocks to implement the primitives can be potentially constructed in any fabrication technology, CMOS included, there is a natural link between delay insensitive paradigm and SET technology as electrons are a natural way to implement the tokens, which constitute the foundation of delay insensitive computation paradigm. In view of this observation this paper presents SET based implementations of this set of building blocks.

To achieve this, delay-insensitive functional blocks were chosen from [9] which together formed a universal class of modules. Utilizing the tunneling effect and the ability to control the individual electrons, we construct topologies such that the movements of a positive charge around the circuit follow certain states, depending on the inputs and input combinations. We then extract these signals and combine them to form the desired output. In this paper we demonstrate that delay-insensitive circuit implementations are possible in SET technology by using a reasonable amount of circuit elements. With the basic blocks we present in this paper one can essentially construct any asynchronous circuits in SET technology. Thus the next step is to construct larger circuits utilizing those blocks and to demonstrate that useful computations can be effectively achieved by combining the delay insensitive paradigm with the single electron tunneling technology.

The paper is organized as follows: In Section 2, the universal set of building blocks is introduced and the functionality of each basic block is described. In Section 3, an implementation in SET circuitry is presented for each basic building block as well as simulation results showing the behavior of the proposed circuit. This is followed by conclusions in Section 4.

II. BASIC BUILDING BLOCKS FOR DIC

The behavior of the universal set of DI primitives can be better described in terms of inputs and outputs *transitions* rather than in terms of absolute logic levels as it is common in conventional digital logic. The DI primitives are as follows:

- Wire(a ; b) - A transition of *a* causes a transition of *b* after an arbitrary amount of time;
- Fork(a ; b,c) - a transition of *a* causes a transition of *b* and *c*, both independent of each other regarding the timeframe;

S. Safiruddin and S.D. Cotofana are with the Computer Eng. Group in TU Delft, Mekelweg 4 2628 CD, Delft, The Netherlands. (Corresponding author e-mail: s.safiruddin@ewi.tudelft.nl)

1-4244-0608-0/07/$20.00 © 2007 IEEE.

- Merge(a,b ; c) - functions as a conventional XOR gate, as a transition of either *a* or *b* causes a transition of *c*.
- C-Element(a,b ; c) - is a state holding element. It waits until both *a* and *b* have undergone a transition before causing a transition of *c*;
- Sequencer(a,b,c ; ao,bo) – here *a* and *b* are inputs and *c* is the control. A transition of *a* and *c*, not necessarily in that order, causes *ao*, and a transition of *b* and *c* causes *bo*. A transition of *a* and *b* and then of *c* causes a transition of *ao* or *bo*, chosen arbitrarily, and a further transition of *c* causes the transition of *ao* or *bo* depending whether *ao* or *bo* transitioned previously, respectively.
- Tria(a,b,c ; d,e,f) – is a further extension with similar functionality where *a* and *b* cause *d, b* and *c* cause *e* and *a* and *c* cause *f*;
- Toggle(a ; b,c) - functions like the conventional toggle flip-flop. A transition on *a* causes a transition on *b* and the next transition of *a* causes a transition on *c* and so on.

In [9] and [10] it was demonstrated that this is a universal set and that logic gates and memory elements can be implemented only using such DI primitives. The aim of this paper is to present the implementation of each one of these basic building blocks in single electron tunneling technology.

III. IMPLEMENTATIONS

Before presenting the proposed SET implementations we would like to explain the notations and conventions that we're using in the discussion:

1. An input signal designated with an appended '^' means that an inverted signal is used, for example Va^. This inverted signal is produced with an inverting buffer being applied to the original signal, as proposed in [4] and graphically depicted in Figure 1.
2. All tunneling junctions have a resistance of 25.8kOhms. When a tunnel junctions appears in the figure it is designated with a J and a number, and when the circuit parameters are described the capacitance of the tunnel junction is referred to with a C and the same number.
3. The power supply voltage is 16mV and logic '1' is encoded as 16mV and logic '0' as 0 mV.
4. All the presented simulations were done using SIMON 2.0 software [8].

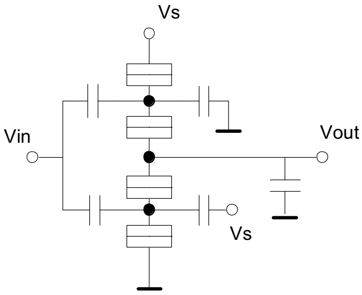

Figure 1. Static Inverting Buffer

A. MERGE

Figure 2 presents a SET circuit topology which provides the required behavior for a Merge. The input signals are Va and Vb and the output signal is Vc. Since the functionality is equivalent to that of an XOR gate an AND-OR-INVERT implementation is the most appropriate. The AND gate and the OR gate are built by utilizing the generic design of the linear threshold gate described in [5]. The circuit comprises of 7 tunneling junctions and 17 capacitors. The circuit parameters correspond to those in [5], specifically, Ca = Cb = Ct = 0.5aF; Cs1 = 9.5aF; Cs2 = 10.5aF; Cg = 10aF; C1 = C2 = C3 = 0.1aF. The inverter is implemented using the inverting buffer in [5].

The circuit operates as follows. When both inputs are low the inverted input signals are high causing an electron to tunnel through J2 leaving a positive charge on n2. This in turn causes an electron to tunnel through J3 leaving a positive charge on n3. This is inverted and so the output becomes low, as it should be. If one of the input signals undergoes a transition then the voltage of J2 (and J1) becomes lower than the critical voltage causing the electron to tunnel back leaving no charge on n2 (and n1). This reduces the voltage over J3 to under the critical voltage and the electron tunnels back leaving no charge on n3. This value is complemented afterwards by the output inverter thus the output becomes high, as it should. If the second signal also undergoes a transition then the voltage over J1 becomes higher than the critical voltage causing an electron to tunnel. This cause an electron to tunnel through J3 leaving a positive charge on n3 corresponding to a low output, as it should. If subsequently one of the signals transitions again, going low this time, the circuit goes into the previous state with no charges on n1, n2 and n3. If after that the other signal also transitions the circuit returns to the original charge neutral state.

The simulation results for the circuit topology presented in Figure 2 are displayed in Figure 3. One can easily observe that they match the expected behavior of a MERGE block.

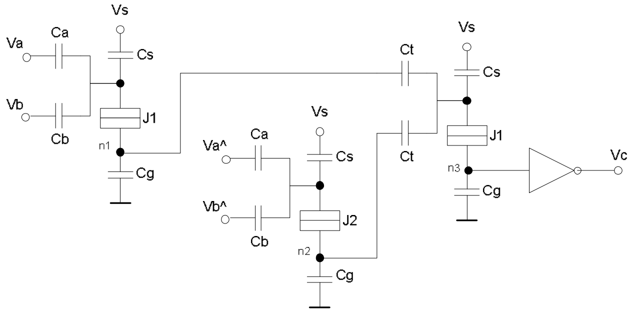

Figure 2. Merge Circuit Diagram.

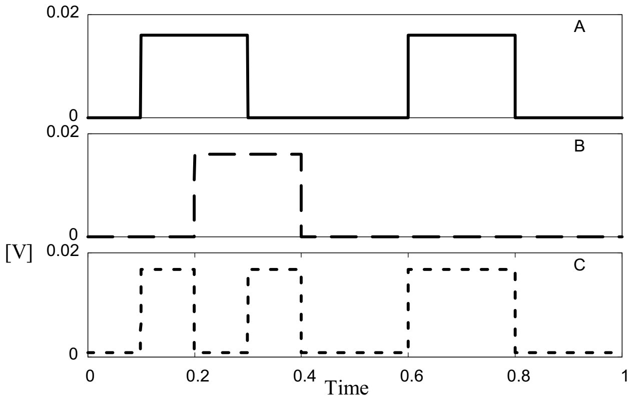

Figure 3. Simulation results for the Merge

B. C-ELEMENT

Figure 4 presents a SET circuit topology that is designed to provide the required behavior for a C-Element. The input signals are Va and Vb and the output signal is Vc. The circuit comprises of 2 tunneling junctions and 4 capacitors. The circuit parameters are as follows: Ca = Cb = 3aF; Cs = Cg = C1 = 10aF; C2 = 0.1aF.

The circuit operates as follows. If both input signals are low the circuit is in its initial neutral state and no charge transport occurs. If either one of the input signals undergoes a transition and becomes high the voltage over J2 increases but does not become critical yet. If the second signal also goes high then the voltage over J2 becomes larger than its critical voltage and an electron is transported from n3 to n2. This causes the voltage over J1 to exceed the critical voltage and the electron further tunnels from n2 to n1. The output is now high, as it should as its value is determined by the charge on n3. Subsequently, when either one of the inputs goes low the voltage over J1 decreases, while the voltage over J2 increases but not above the critical voltage. Only when the second signal also goes low does the voltage over J2 exceed the critical voltage and an electron is transported from n2 to n3 leaving n2 positively charged. This charge on n2 causes the voltage over J1 to exceed the critical voltage and the electron residing on n1 tunnels through into n2. Consequently, the circuit is back in its initial neutral state, as it should.

The simulations results can be seen in Figure 5. As it can be observed the proposed SET circuit produces the correct C-Element functionality. The output goes high only when both input signals have gone high, and subsequently goes low only when both input signals have gone low.

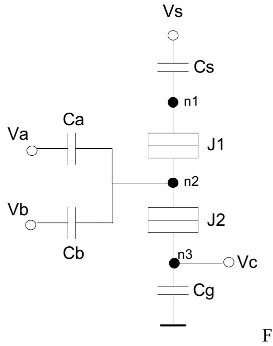

Figure 4. C-Element circuit diagram

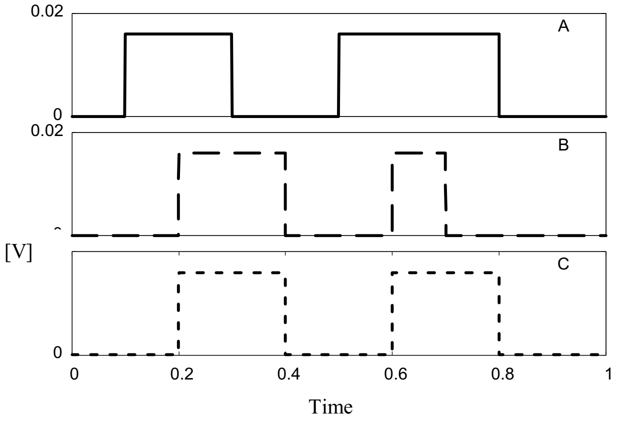

Figure 5. Simulation results for the C-Element

C. SEQUENCER

Figure 6 presents a SET circuit topology which provides the required behavior for a Sequencer. The input signals are Va, Vb, and Vc and the output signals are Vao and Vbo. The circuit comprises of 10 tunneling junctions and 20 capacitors. The circuit parameters are follows: Ca = Cb = Cc = 3aF; C1 = C3 = C5 = C7 = 10aF; C2 = C4 = C6 = C8 = 0.1aF; Cg = Cs = 10aF; Cs1 = Cs2 = 10.5aF; C9 = C10 = 0.1aF; Ct = 0.5aF.

The circuit is actually made up from 4 C-Element structures and operates as follows. With all inputs low the circuit is in a neutral state with no transported charge. If the two inputs A and C go high an electron tunnels through J2 and then through J1 and propagates from node n2 to n1 due to the transport mechanism built in such a C-Element, leaving a positive charge on n2 and a negative one on n1. From here, according to the sequencer specification, either A and C would go low causing the electron to go back to n1 resulting in the neutral state, or B would go high and C would go low meaning C^ going high resulting in an electron tunneling through J4 and J3. This propagation through J4 and J3 is again similar to that of the C-Element mechanism. With n2 positive and the two corresponding inputs high the critical voltage of J4 would be exceeded resulting in an electron from n3 tunneling through J4. The voltage over J3 would exceed the critical voltage due to the positive charge on n2 and the negative charge on the other side and the electron would tunnel to n2 leaving no charge on n2 and a positive one on n3. Subsequently, with inputs A and B high and C low, either A would go low and C^ low, or B would go low and C^ low resulting in the voltage over J8 and J4, respectively, increasing and exceeding the critical voltage. This would cause an electron to tunnel into n3 and increase the voltage over J7 and J3, respectively, above the critical voltage causing an electron to tunnel through J7 and J3 leaving a positive charge on n4 and n3, respectively. An electron effectively propagates from n4 or n2 into n3. From there the aforementioned mechanisms would apply. The net effect is thus that the positive charge moves around the circuit depending on input signal transitions, until it reaches node n1 where the circuits returns its initial neutral state. The nodes n2 and n3 correspond to Ao, which is evaluated by an OR gate implemented using a linear threshold gate as introduced in [5] and taking them as inputs. The nodes n3 and n4 correspond to Bo and are thus the inputs for the OR gate to produce the Bo output signal. If, from the neutral state A and B go high and then C, then one of the propagation routes is chosen arbitrarily and either node n2 or n4 becomes positively charged. Subsequently, the circuit is in the state as described above after the first propagation and the extra signal which went high is still pending for the C to transition.

The simulation results are graphically displayed in Figure 7. As one can observe from the first segment of the simulation, which corresponds to the first A pulse, the basic Sequencer functionality is delivered. Input A going high or low followed by or preceded by input C going high or low produces a toggling of the corresponding output. The same holds for input B. In the second part of the simulation we can see the additional Sequencer functionality showing that if A goes high and then B goes high before C going high two subsequent transitions of C cause a toggling of Ao and Bo outputs in an arbitrary order.

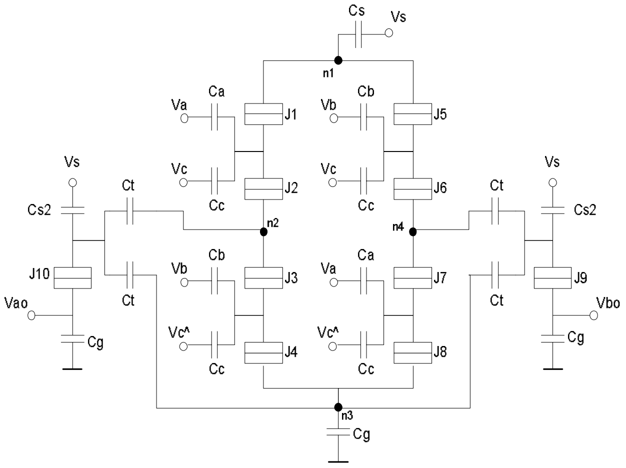

Figure 6. Sequencer circuit diagram

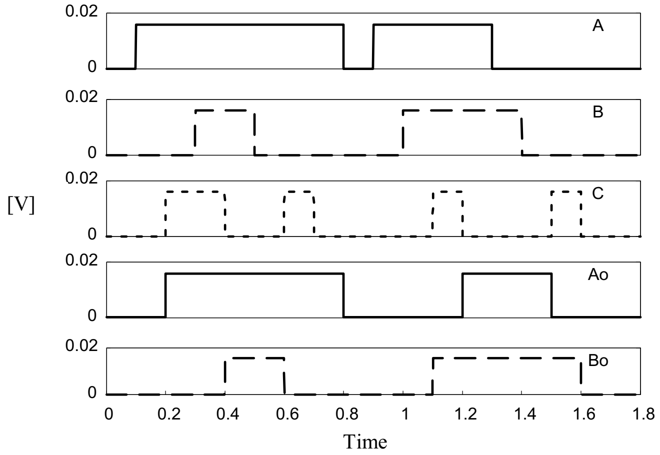

Figure 7. Simulation results for the Sequencer

D. TRIA

Figure 8 presents a SET circuit topology that provides the required behavior for a Tria. The input signals are Va, Vb, and Vc and the output signals are Vd, Ve, and Vf. The circuit comprises of 27 tunneling junctions and 50 capacitors. The circuit parameters are as follows: Ci = 3aF; Cg = Cs = 10aF; C1 = 10aF; C2 = 0.1aF; Ct = 0.21aF; C3 = 0.1aF; Cs2 = 10.5aF.

The circuit operates in a very similar way the Sequencer does. An electron moves to node n1 from n2, n3 or n4 depending on which input signals go high. Only if the same two signals then go low again does the electron propagate back. This is in effect the C-Element functionality. Subsequently, the node which has the positive charge has three nodes from which an electron may potentially arrive, depending on which input signals undergo transitions. Thus, each node has a pathway to other 3 nodes opening up only due to transitions which are possible if that node has been reached. As with the Sequencer, the positive charge moves from node to node depending on the transitions made by the input signals. Each node contributes to one or more of the outputs as follows. n2, n5, n6, and n8 correspond to output D and a 4-input OR gate is used to combine these node outputs to get the output D. Similarly, output E is formed from n3, n5, n7, and n8 and output F is formed from n4, n6, n7, and n8.

The simulation results are displayed in Figure 9. The first two transitions are of A and B going high. An electron thus propagates from n2 to n1, and the charge on n2 signals the output D. Subsequently, B goes low and C goes high causing the electron to propagate from n5 to n2. The charge on n5 thus signals the output E, however, the charge on n2 is now zero but the output hasn't changed, it has been moved to node n5. Node n5 thus signals both output D and E. The rest of the simulation proceeds in a similar fashion and clearly indicates that the topology in Figure 8 is able to deliver the Tria functionality.

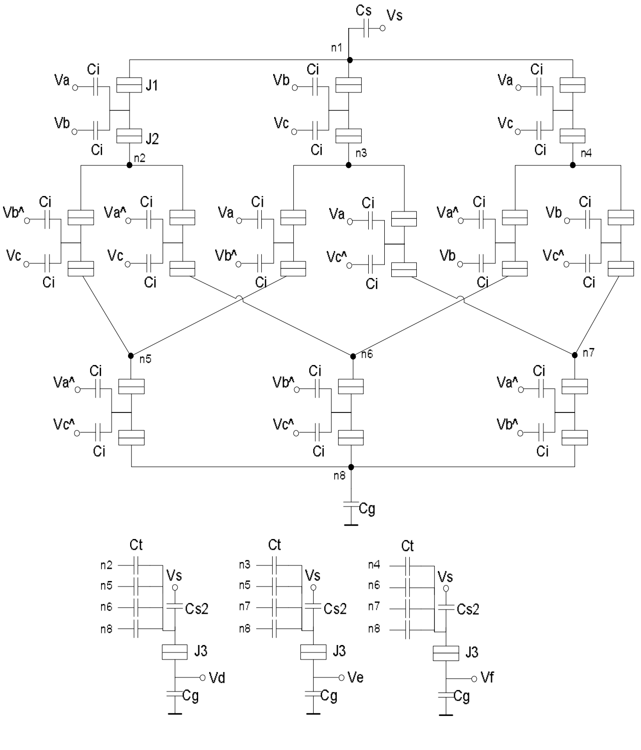

Figure 8. Tria circuit diagram

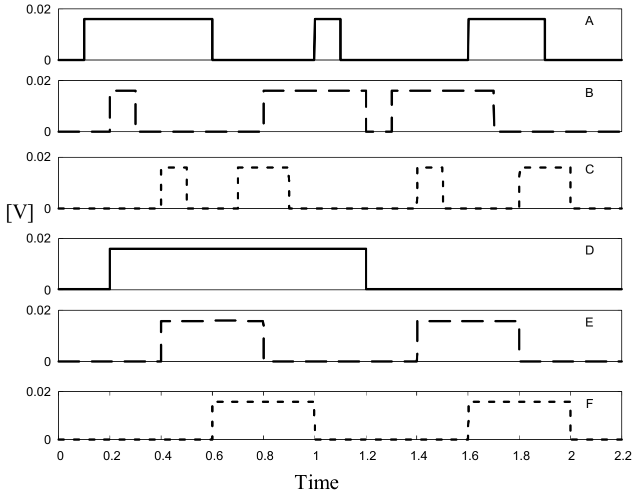

Figure 9. Simulation results for the Tria

E. TOGGLE

Figure 10 presents a SET circuit topology which provides the required behavior for a Toggle. The input signal is Va and the output signals are Vb and Vc. The circuit comprises of 10 tunneling junctions and 17 capacitors with the following values: Ca = 4aF; Cr = 3aF; Cb = 3.6aF; Cg1 = 11aF; Cg2 = 13aF; Cg3 = 18aF; Cs = Cg = 10aF; Cs2 = 10.5aF; Ct = 0.5aF; C1 = C3 = C5 = C7 = 10aF; C2 = C4 = C6 = C8 = 0.1aF.

The circuit operates as follows. It starts in a neutral state with the input low and with no charge transported. If Va goes high an electron tunnels through J2 from node n2 to node n5 and raises the voltage over J1 above J1's critical voltage. The electron then tunnels further to n1 through J1, leaving a positive charge on n2. The electron doesn't return if Va goes low, as the critical voltage of either J1 or J2 is not exceeded. When Va goes low, Va^ goes high and an electron is transferred from node n3 to n2 by tunneling through J4 and then J3. With a positive charge now on n3 if Va goes high again an electron tunnels through J6 and then J5 from n4 to n3. Finally if Va goes low again, Va^ going high causes the voltage over J8 to exceed its critical voltage causing the electron residing on n1 to tunnel through J8 and then through J7 to neutralize the positive charge on n4 and bring the circuit back to its initial neutral state. The charge on node n2 corresponds to the first pulse of Va and the charge on node n3 corresponds to the signal till the second pulse, and therefore these two nodes contribute together to the first output of the Toggle, Vb. These nodes are therefore combined using a linear threshold OR gate as introduced in [5]. Nodes n3 and n4 determine the value of the Vc signal thus they are also combined using an OR gate to evaluate Vc.

The simulation results are displayed in Figure 11. As is expected, output B toggles for each positive edge of the input and output C toggles for each negative edge of the input. During the first pulse only B is high which corresponds to node n2 and after the end of the first pulse both B and C are high corresponding to node n3. This is as expected as n3 supplies the output for both B and C. During the second pulse only C is high, corresponding to node n4. After the second pulse the circuit is in its initial neutral state again and none of the outputs are high. The cycle then repeats itself.

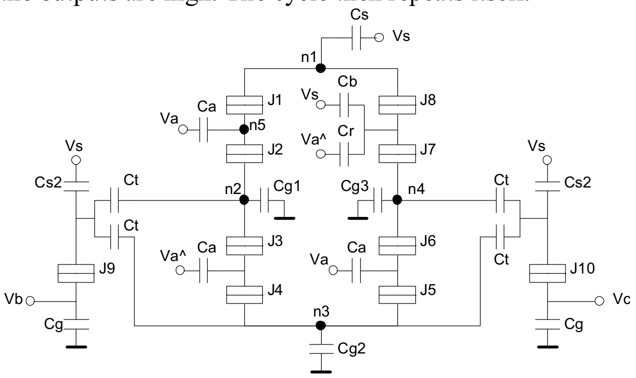

Figure 10. Toggle circuit diagram

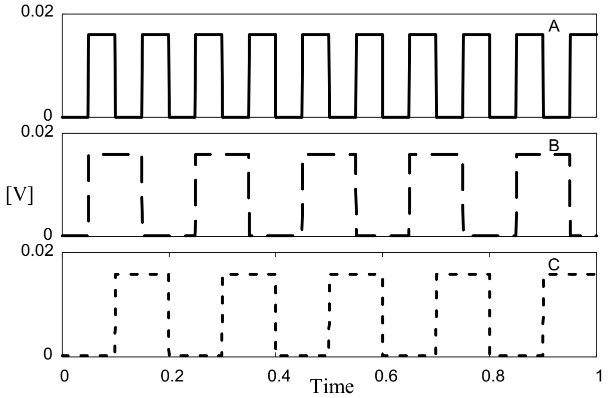

Figure 11. Simulation results for the Toggle

IV. CONCLUSION

We have presented SET based implementations of building blocks corresponding to Delay Insensitive (DI) primitives. The proposed circuits were validated by means of simulations and they are able to deliver the required behaviors. As future work we plan to demonstrate how one can utilize those blocks to create any desired functionality using DI paradigm. We expect this to require some buffering techniques as the topologies we propose are mainly passive circuits. Such active buffers have been already proposed for SET based gates but we believe that they will require some adaptations in order to be utilizable in the context of DI circuits. The presented circuits function in ideal conditions of 0K with no co-tunneling or background charge effects assumed. It would be interesting to develop circuits that could be robust enough to withstand higher temperatures as well as to withstand interference due to random quantum effects. Additionally, in [9] a novel set of primitive modules was proposed where the interconnection lines can be bi-directional and buffered. This consists of a Fork(a; b,c) as previously described, P-Merge(a,b; c), S-Join(a,b,c ; a,b,c), IOM(a,b ; b,c) and an R-ATS(R,T ; R,T). The realization of designs for those bi-directional blocks would broaden the implementation possibilities even further and constitute the subject of future work.

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, S. Cotofana, and S. Vassiliadis, "Static Buffered SET Based Logic Gates", in *Proceedings of the 2nd IEEE International Conference on Nanotechnology (NANO),* pages 491 – 494 (Arlington, USA), August 2002.

[5] C. Lageweg, S. Cotofana, and S. Vassiliadis, "A Linear Threshold Gate Implementation in Single Electron Technology", in *Proceedings of the IEEE Computer Society Workshop on VLSI,* pages 93 – 98 (Orlando, USA), April 2001.

[6] 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.

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

[8] 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}:934-944, September 1997. http://www.lybrary.com/simon/

[9] J. Lee, F. Peper, S. Adachi and K. Morita, "Universal Delay-Insensitive Circuits with Bidirectional and Buffering Lines" IEEE Transactions on Computers, vol. 53, no. 8, August 2004.

[10] P.G. Lucassen, "A Denotational Model and Composition Theorems for a Calculus of Delay-Insensitive Specifications", PhD. Thesis, Rijksuniversiteit Groningen, 1994.