Stigmergic Search with Single Electron Tunneling Technology Based Memory Enhanced Hubs

Saleh Safiruddin and Sorin Cotofana, Senior Member IEEE
Faculty of EE, Mathematics and CS
Delft University of Technology, Delft, The Netherlands

Ferdinand Peper, Member IEEE
Brain ICT Laboratory
National Institute of Information and Communications Technology, Kobe, Japan

***Abstract*—Fluctuations have recently been recognized as powerful resources that can be exploited to drive computations, but their use has mostly been limited to logic circuits. This paper goes further and explores a more general framework, in which computation is modeled as a process with a multitude of fluctuating tokens that interact with each other directly or via stigmergy. For the implementation of these concepts Single Electron Tunneling (SET) technology is a strong candidate, since it combines a key element of fluctuation-driven systems, i.e., fluctuating tokens, with the potential for manufacturing in traditional materials (silicon) as well as alternatives, such as molecules. We propose computational elements, i.e., Memory Enhanced Hubs (MEHs), that contain functionality to pass fluctuating signals through them, as well as stigmergic functionality to store a state temporarily and reset it. We introduce a SET based design of such a memory enhance hub instance and demonstrate by means of simulations that it function correctly and that MEHs networks operating according to the stigmergic paradigm can be constructed.**

I. Introduction

Fluctuations have traditionally been considered impediments to the operation of systems and circuits. The ever-decreasing Signal/Noise ratios in integrated circuitry call for strategies that go beyond the conventional methods, which amount to suppressing noise or adding redundancy to systems. One such strategy aims to find solutions to computation problems through the use of stochastic search, but other strategies have also been proposed [1], such as the encoding and transmission of information by signals that are based on noise, the enhancement of signals through stochastic resonance, and the use of noise to synchronize elements in distributed systems. Stochastic search is a powerful method, and it is no surprise that it is employed in biological systems [2]. When used in organisms, it brings together agents so that they can react with each other. Importantly, this mechanism requires few resources, since it is driven by fluctuations, which come for free. In an optimization context the use of stochastic search has been proposed in [3], whereby fluctuations of electrons in a Single Electron Tunneling (SET) system drive the search of a simulated annealing algorithm. A more computationally oriented use of stochastic search is found in Brownian circuits [4], [5], in which tokens find their way on a circuits computational paths through the use of fluctuations; this method allows tokens to backtrack out of deadlocks through fluctuations.

The above are quite traditional frameworks in which stochastic search is applied, and, though quite different, they have in common the simplicity of interactions between tokens or processes that drive the search. Interactions between tokens in Brownian circuits, for example, are restricted to take place in the inside of circuit elements, and even if we allow more flexibility by using repulsion between tokens implemented as electrons [6], the model is quite limited. More complex interactions require added functionality, but in a nano-architecture context, it is difficult to implement such functionality in just tokens. Enter *stigmergy*, which is a well-known mechanism in swarm-based systems. It amounts to the interaction between distributed elements through the use of markers ("bread crumbs") that, when deposited in the environment by the elements, influence the behavior of elements running into the markers. This indirect way of interaction keeps the elements simple, while allowing quite complex phenomena to be modeled. Nest building by social wasps, for example, is based on stigmergy [7]: rather than remembering the total architecture of a nest, each wasp possesses a set of rules, of which the application at local scales lead to a nest built as if it was designed in a top-down way.

This paper examines the use of stigmergy in a model that is driven by fluctuating tokens. We propose a building block that lets fluctuating tokens pass through under certain conditions, yet blocks tokens under different circumstances, while changing the local state of the building block. Simulations of the building block implemented by SET technology confirm that it works as designed. The paper is organized as follows. Section 2 introduces the new paradigm in which stigmergy is used in combination with stochastic search by fluctuating tokens. Section 3 discusses the implementation of building blocks in SET technology, and shows simulation results of them. Section 4 discusses how the building blocks can be combined into bigger systems. We finish with conclusions in Section 5.

II. STIGMERGIC COMPUTATION ELEMENTS

Most stigmergic optimization algorithms in literature use the ant-pheromone model. In this model, the different solutions to an optimization problem are encoded as trails in a network that can be followed by ants. A trail that is favored by ants will then correspond with a solution found by the algorithm. Ants are known to use pheromones to mark a trail, which conveys the message to subsequent ants that the trail is a favored one, and thus should be followed. Still, this strategy may cause convergence to local optima, because once ants follow a certain trail, all will go there. Ants have thought up a smart way to avoid this problem. First of all, there is some randomness in their behavior, to ensure that not every ant goes on the same trail, but rather that there is some variety. Second, pheromones evaporate, which imposes some time limit on the positive bias of trails. In this way infrequently used trails will retain their pheromones to a lesser extent than the more popular ones. As a result of this mechanism, the algorithms will gradually develop some favored trails, and these tend to correspond to optimal (or almost optimal) solutions.

In the implementation of these ideas, we will use an element, called Hub, that is used to transfer signals, without there being a bias in any direction. Signals in this context are modeled as tokens, which are discrete units that cannot be split or be unified with other tokens. Tokens thus serve as a kind of particle, and in a SET context it makes sense to use electrons for this. The discrete nature of tokens makes them ideal candidates to model ants. Figure 1 presents a Hub [5] schematic with the possible ways a token can go on it.

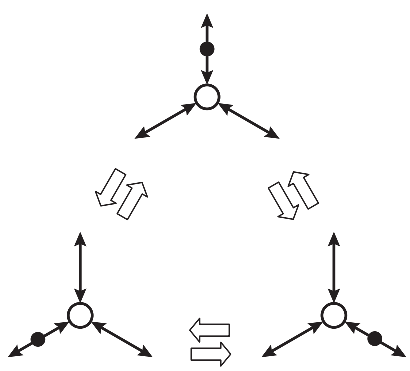

Fig. 1. Hub element, which is indicated by an open circle with three bidirectional wires leading to it. A token (indicated by a black blob) is allowed to fluctuate among these wires freely.

Since the Hub has node degree 3, it cannot used as is in expressing graphs with nodes of arbitrary degree, but fortunately it is possible to use an encoding like in Figure 2.

If Hubs are allowed with larger degrees, it will be possible to make more direct encodings of graphs. Depending on technology, however, such Hubs may not be feasible, so we assume in the following discussion the minimal case in which Hubs have degree 3.

The encoding of trails and pheromones requires a variation of the Hub, in which a memory is added to store whether there is a pheromone present at the Hub or not. For this purpose we use a Hub at which one of the lines contains a memory that ”catches” a token that happens to swerve to that line. This line is called the *Memory Line* and the new Hub is called a *Memory Enhanced Hub*. When a token is caught by a memory, it changes the state of the memory, making it impossible for subsequent tokens to enter the Memory Line. Only if the memory is reset will it be possible for a subsequent token to enter the Memory Line, and being caught. So, if a memory is set, tokens can freely move between the remaining two lines of a Memory Enhanced Hub, but in an unset memory, there is a significant probability that the token will be moved to the memory. By connecting Memory Enhanced Hubs in a serial way such that the normal lines, i.e., non-Memory Lines, are connected to each other (Figure 3), it becomes possible to form trails in an ant-inspired algorithm. The Memory Lines in this model then only serve as devices to catch tokens passing by and disposing of them afterwards. If the sequence of Memory Enhanced Hubs in unset states is long enough, the probability that a token can pass through it without being caught becomes increasingly small. Of course, by being caught on a Memory Line, the token increases the probability that subsequent tokens can pass through the sequence of Memory Enhanced Hubs, and thereby increases the strength of the corresponding trail.

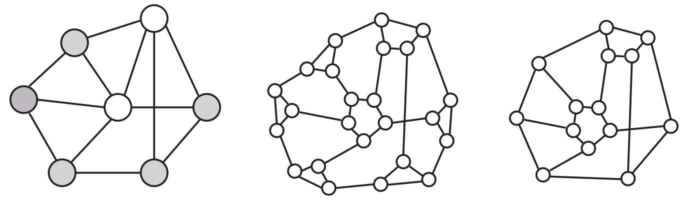

Fig. 2. Graph (left) and its encoding in terms of Hubs (center). This encoding graph can be simplified by encoding nodes with degree 3 directly by Hubs (right). The nodes that are originally of degree 3 are colored gray.

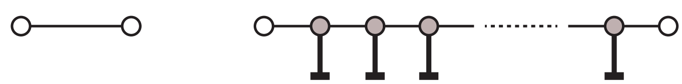

Fig. 3. Connection between two Hubs (left), and its encoding by a sequence of Memory Enhanced Hubs (right), which are indicated by the color gray. The Memory Lines are thicker than regular lines and they end up in terminators, which serve as ground to dispose of tokens when memory is reset.

By resetting a memory line of a Memory Enhanced Hub in a sequence, on the other hand, we decrease the strength of the trail. This corresponds to evaporating a pheromone, and it can be implemented by adopting a stochastic scheme according to which memories are reset. When many tokens pass through a trail, it is likely that most memories of the Memory Enhanced Hubs on the trail will be set, increasing the probability that subsequent tokens can pass through unabated. Since resetting of memories takes place at the same pace everywhere, the trails least visited by tokens will have more unset memories on average than more popular trails, and most tokens attempting to go through that trail will be caught. This provides us with a natural way to implement ant-based stigmergic algorithms in a SET framework. In the next sections we introduce the details of such an implementation, and present some simulation results.

III. SET Based Memory Enhanced Hub

A. Concept

We consider a search space with tokens navigating through interconnected nodes. To direct the search, make it more efficient, or make a self-organizing system, tokens can communicate messages for each other. We focus on indirect communication, where tokens can leave messages, or "bread crumbs" for other tokens. Based on those messages subsequent tokens passing through the same space again are then redirected in some specific way. We propose extending the Hub functionality to enable tokens to leave a certain message as it passes through the Hub. The token, in effect, changes the functionality of the Hub as it passes through. The change itself may be very simplistic, however, a self-organization can emerge. In SET circuits, tokens are represented by electrons, and thus are indistinguishable from each other. Thus we focus on messages for any token, instead of messages for specific types of tokens.

B. Extended Hub Functionality

In the original Hub functionality, previously implemented in SET in [8], where three channels are connected to a node, tokens enter one channel and randomly exit from one channel, which could also be the one it just came from. To provide support for stigmergic search the Hub functionality has to be extended such that it does not have a static behavior anymore, but it is adaptable. Once a token passes through, the Hub should react in a predetermined way and reconfigure its behavior. Specifically, this means changing the way tokens are allowed to move through the channels of the Hub. In this line of reasoning the Hub has to be augmented with some memory facility. Furthermore, to enable the construction of rectangular Hub meshes we focus the description on a Hub with 4 channels instead of 3. We note however that, at least theoretically speaking, Hubs with more than 4 channels can be developed.

C. Channel Routing

A message can be left in a Hub, to change the channel routing. There are a number of ways the channel routing can be adapted as follows:

1) Token entering channel
   - Channel block bi-directional once a token enters through a channel, that channel is blocked as depicted in Figure 4.
   - Channel block uni-directional once a token enters through a channel, that channel is blocked in one direction, acting like a valve. The two possible situations are presented in Figure 5 and Figure 6
2) Token exiting channel
   - Channel block bi-directional: once a token exits through a channel, that channel is blocked as presented in Figure 7
   - Channel block uni-directional: once a token exits through a channel, that channel is blocked in one direction, acting like a valve. The two possible situations are presented in Figure 8 and Figure 9.

By using combinations of these options on certain channels we can create various adaptable Hub functionalities.

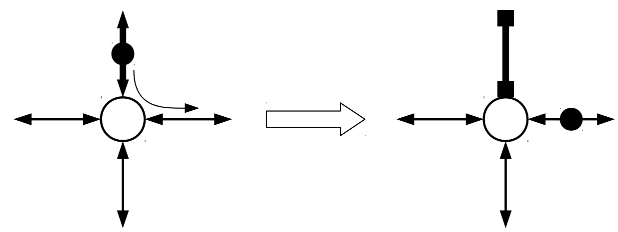

Fig. 4. Token entering the Hub through a channel blocks that channel.

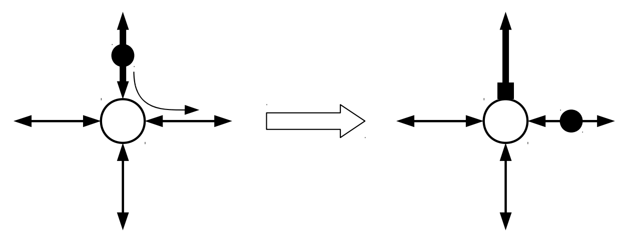

Fig. 5. Token entering the Hub through a channel blocks further tokens entering through that channel.

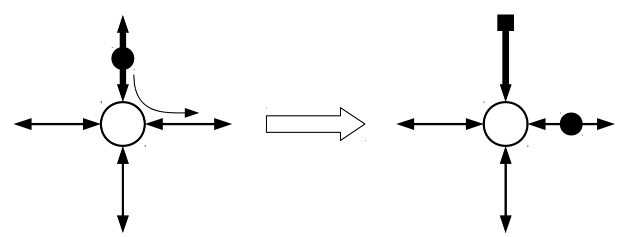

Fig. 6. Token entering the Hub through a channel blocks further tokens exiting through that channel.

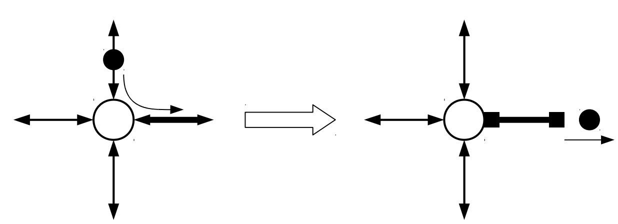

Fig. 7. Token exiting the Hub through a channel blocks that channel.

D. SET Background

In classical physics, an electron was treated as a particle and modeled as a moving charge. It was discovered however,

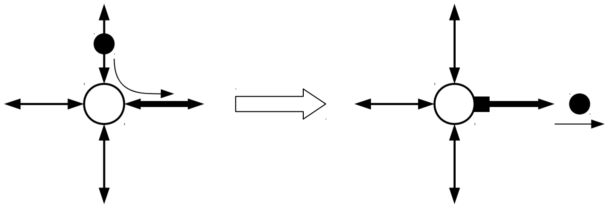

Fig. 8. Token exiting the Hub through a channel blocks further tokens entering through that channel.

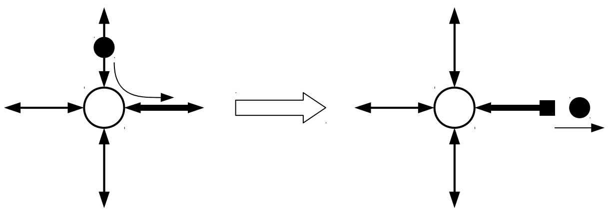

Fig. 9. Token exiting the Hub through a channel blocks further tokens exiting through that channel.

that electrons sometimes behave as waves, and in quantum mechanics, electrons are modeled as such. This means that there is a non-zero probability that an electron will tunnel through a potential barrier, if the electron's final energy state is lower behind the barrier. This phenomenon is called quantum tunnelling and it is the basis of Single Electron Tunnelling technology. The junction through which electrons tunnel is called a tunnel junction and it is a new circuit element, depicted in Figure 10. Tunnel junctions together with capacitors are the basic building blocks in SET based circuits [9], [10].

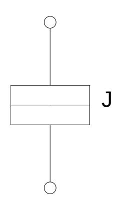

Fig. 10. Circuit symbol for SET junction

Tunneling through a junction becomes possible when the junction's actual voltage V_j exceeds the junction's critical voltage $V_c = \frac{q_e}{2(C_j+C_e)}$, where $q_e = 1.602 \cdot 10^{-19}C$, the capacitance of the junction is C_j, and the capacitance of the remainder of the circuit as seen from the junction is C_e. The delay of such circuits cannot be analyzed in the traditional sense. Instead, for each transported electron one can describe the switching delay as $t_d = \frac{-\log P_{\text{Error}} R_t}{|V_j|-V_c}$, where R_t is the junction resistance and P_{Error} is the probability that the desired charge transport has not occurred after t_d seconds. This probabilistic delay complicates the direct utilization of SET-based computation in building synchronously timed arithmetic units and since the exact tunnel time of an electron is not known, extensive error correcting schemes are required and switching times have to be lengthy. This has motivated research for SET architectures and circuits based on delay-insensitive computations so that the problems arising from unknown delays are eliminated altogether [11].

We start with general conditions which have to be met to enable single electron manipulations. Two major effects have to be considered, namely quantum fluctuations and thermal energy. If either of these two effects is dominant then the electrons are not localized on islands and the desired functionality cannot be achieved.

Since electron tunneling is a quantum mechanical process, an electron's wave function extends through potential barriers, and the electron is spread over the islands in a SET circuit. If this effect was to prevail there would be no localized charges and computations using electrons would not be possible. To ensure that the charge of an electron is quantized on each specific island the tunnel junctions must have a sufficiently high tunneling resistance, so that the charging energy, also called the Coulomb energy, dominates over the quantum charge fluctuations. This can be expressed as:

$$\frac{{q_e}^2}{2 \cdot C_j} \cdot R_j \cdot C_j >> h => R_j >> h/{q_e}^2 = 25.8k\Omega$$

where h is Planck's constant, C_j is the tunnel capacitance, and R_j is the tunneling resistance.

The second effect that has to be considered is that of thermal energy. If the thermal energy dominates over the charging energy, E_c, the quantization effects become again non-observable. The condition is then $E_c = \frac{{q_e}^2}{2 \cdot C} > k_B \cdot T$, where k_B is Boltzmann's constant and T is the absolute temperature.

In the last few years there has been considerable interest in researching methods to effectively utilize the basic SET properties. Such efforts include [12], [13], [14], [15], [16], which are based on Single Electron Encoded Logic and Electron Counting concepts. Theoretical results on the area and delay complexity of arithmetic operations using those new paradigms indicate great potential. However, one of the most important challenges for implementing circuits based on quantum tunneling thus far has been the stochastic nature of the tunneling process, which in the context of this paper it is a paradigm enabler factor.

E. Memory Enhanced Hub Implementation Example

In this section we focus on a SET technology implementation of a Memory Enhanced Hub instance, which on one of its channels supports the policy that in the case a token is exiting the Hub through that channel it blocks further tokens from exiting that channel, depicted in Figure 9. All the other cases can be implemented in a similar fashion.

In the following implementation, all tunneling junctions have a resistance of 100kΩ, which has also been used in previous studies, [12], [13], [14], [15], [16]. The source voltage V_s is set at $16mV$. All simulations were done using SIMON 2.0 software [17] and were performed at $0.5K$ temperature. The temperature of $0.5K$ was chosen to demonstrate the coexistence of fluctations of quantized electron charges and deterministic behaviour in a circuit. As was previously mentioned,

TABLE I
CIRCUIT PARAMETERS.

Parameter	Value (aF)
J_2, J_4, J_6, J_8, J_{10}, J_{12}, J_{14}, J_{16}, J_{18}, J_{20}, J_{22}, J_{24}	0.1
J_{15}	4.9
J_3, J_5, J_{17}, J_{19}, J_{21}, J_{23}	9.5
J_1, J_7, J_9, J_{11}, J_{13}	10.0
C_{g1}, C_{g6}, C_{g10}	10.0
C_{g2}, C_{g4}, C_{g5} , C_{g7}	10.1
C_{g3}, C_{g8}	10.2
C_{g9}	11.0
C_{s4}, C_{s5}, C_{s6}, C_{s7}, C_{s10}, C_{s11}	0.1
C_{s2}, C_{s3}, C_{s8}, C_{n1}, C_{n2}	0.2
C_{s9}	0.3
C_{s1}	1.0

the thermal energy should not dominate the circuit elements, however, to achieve the desired Hub functionality, there should be enough thermal energy flucutations to enable electrons to tunnel from one island to another. This was achieved by bringing the voltages over certain tunneling junctions close enough to their critical voltages such that an increase in the energy of an electron at the junction due to the extra thermal energy would at random moments cause the electron to tunnel forward, and when the thermal energy decreases to tunnel back again. The thermal energy is thus effectively used as a random control voltage. The capacitance of an island should therefore not exceed $463aF$ for a temperature of $0.5K$.

Figure 11 presents a SET circuit topology that exhibits the proposed Hub functionality in Figure 5. The circuit parameters are presented in Table I.

The circuits consists of a token supply, the blockable output of the Hub, the other outputs of the Hub, and the Hub memory. In this case, we have only one Hub channel which is blockable. However, Hubs with more than one blockable channel can be directly constructed based on the proposed implementation.

In the token supply, tokens are injected into the circuit from the source for the purpose of simulation. V_a generates the token supply. Here the electron (holes) are injected into the circuit. At each pulse one new electron hole enters the circuit and does not tunnel back to the source. At the end of the blockable channel V_b controls a token sink which is also included for simulation purposes. A V_b pulse consumes a token so that the effect of subsequent token injection can be observed. In a multi-hub circuit this is not required as neighboring hubs can consume the token. Out_A (n_7) is the blockable channel output of the Hub. Out_B (n_4) and Out_C (n_5) are the other output channels of the Hub. In the memory circuit $Memory$ (n_{10}) holds the state of the memory element, which enables/disables the blocking mechanism.

The blocked channel can be unblocked either stochastically, or by a returning token. This can be enabled by connecting the reset output in the memory state circuit either to a constant biasing voltage, or to the path of a returning token.

The circuit operates as follows. Whenever node n_1 has no charge on it, electrons tunnel through junctions J_2 and J_1 to the source resulting in a constant postive charge on the node. Subsequently, node n_2 also has a positive charge resulting from electrons tunneling through junctions J_4 and J_3. This positive charge on n_2 is always replenished soon after it is lost. Once V_a goes high, the charge on n_2 is transferred to n_3 by an electron tunneling through J_5 and J_6 (the charge on n_2 is then replenished).

From n_3 the charge can be randomly transferred to n_4, n_5 or n_6 through the junctions J_7-J_8, J_9-J_{10} or J_{11}-J_{12}, respectively. If it is transferred to either n_4 or n_5 then the charge can fluctuate back to n_3 and forth again.

If it is transferred to n_6 then the charge further tunnels through J_{13} and J_{14} onto node n_7. The charge on n_7 causes an electron to tunnel through J_{17} and J_{18} from n_8 to n_{11} by way of C_{n1}. This charge is subsequently transferred to n_{10} through J_{19} and J_{20}. The charge on n_{n10} raises the potential on n_6 thus blocking any further charges arriving on it.

If $ResetMemory$ goes high, the charge on n_{10} transfers to n_9 through J_{21} and J_{22}. Finally, the electron residing on n_{11} tunnels through J_{23} and J_{24} onto n_9 cancelling out the charge. The increased potential on n_6 is removed allowing charges to be transferred onto node n_6 again.

To demonstrate that the proposed design functions as it should we present the results of a SIMON simulation in Figure 12. We note that the simulations are event-based and thus there is no unit of time, the x-axis merely denotes relative progress in time. The simulation starts with a token injected at the token supply with V_a going 'high' at time 0.1. The charge first fluctuates between Out_B and Out_C. Once the charges reaches Out_A it is trapped there and does not return. This triggers the memory state and $Memory$ goes 'high'. At time 0.5 V_b pulses and the token is consumed with Out_A going 'low'. Next, V_a pulses again at time 0.6, to inject a second token into the circuit. Now we observe that the token only fluctuates between Out_B and Out_C and the output to Out_A is blocked, as it should.

IV. EVALUATION WITH HUB MESH NETWORK EXAMPLE

To demonstrate the application of a Memory Enhanced Hub in a useful circuit, we take an example with 4 Hubs arranged in a 2D lattice structure as shown in Figure 13(1.). For the demonstration, only one channel of one Hub is made blockable, however, any amount of blockable channels can be implemented, in any amount of Hubs, thus theoretically speaking a 2D lattice of any size and with different types of memory enhanced hubs can be constructed. Figure 13 depicts the four stages of the implemented arrangement is going through and the simulation results are presented in Figure 14. In stage 1 a token is injected into the mesh. The token fluctuates in the circuit until it arrives at the blockable channel. In stage 2 the token travels through the blockable channel, leaving it blocked. A second token is then injected into the circuit, which fluctuates in the circuit. In stage 3 the second token reaches the blocked channel but from the reverse direction of the block. In stage 4 the second token passed through the blocked channel causing it to be reset to it's original state and unblocking it. The token then fluctuates through the circuit.

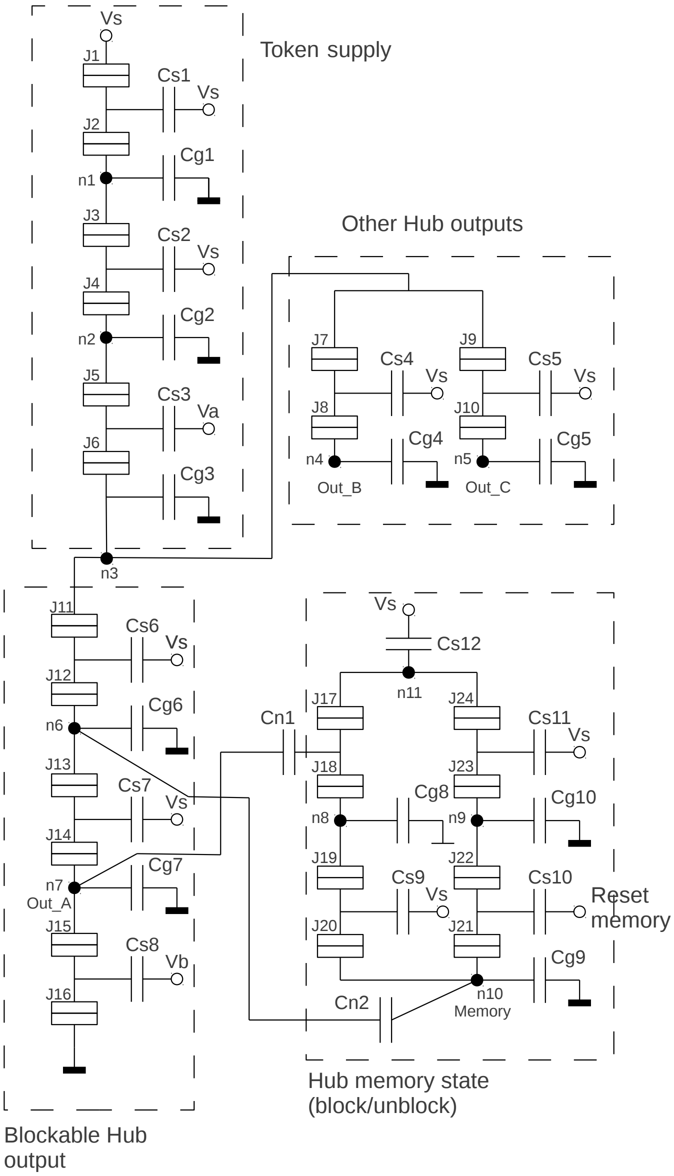

Fig. 11. SET implementation of Memory Enhanced Hub.

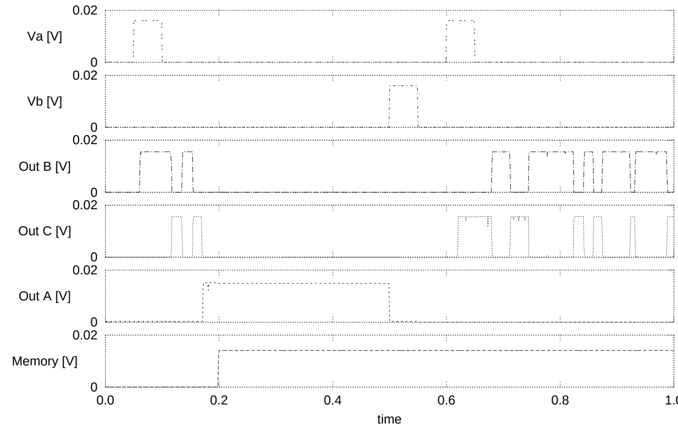

Fig. 12. Simulation results for Memory Enhanced Hub.

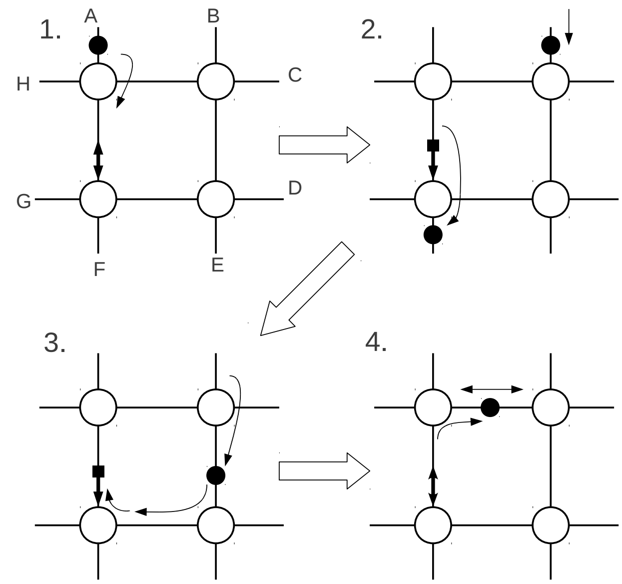

Fig. 13. Memory Enhanced Hub network example: rectangular mesh. An injected token blocks one channel of a Memory Enhanced Hub, which is subsequently unblocked by a second injected token.

V. Conclusion

In this paper we have presented a general framework for stigmergic computation utilizing fluctuating tokens and demonstrated that SET technology can be used for stigmergic computation. We have proposed a novel computational element, a Memory Enhanced Hub, with 6 different possibilities of blocking certain Hub channels based on the fact that: (i) A token entering into or exiting from a Hub channel can block that channel bi-directionally; and (ii) A token entering into or exiting from a Hub channel can block that channel in either direction leaving the channel functioning like a valve. These Hub enhancements allow temporary storage of state which enables stigmergic functionality. We have demonstrated that this new element can be implemented in SET technology by proposing a SET circuit topology which has the functionality of blocking the channel bi-directionally when a token enters that channel and we have verified the circuit by means of simulation. Furthermore, we demonstrated that potentially speaking Memory Enhanced Hubs can be combined in larger networks able to support the stigmergic paradigm by implementing and simulating a small hub network. Our investigation indicates that SET technology is a promising candidate for practical implementations of token based stigmergic computation.

References

[1] F.Peper, L.B.Kish, K.Leibnitz, and J.-Q.Liu, "Methods to exploit noise in the design of complex systems," in *SICE*, 2008, pp. 231–236.

[2] T.Yanagida, "Fluctuations as a tool of biological molecular machines," in *Biosystems 93 (1-2)*, 2008, pp. 3–7.

[3] T.Yamada, M.Akazawa, T.Asai, and Y.Amemiya, "Boltzmann machine neural network devices using single-electron tunneling," in *Nanotechnology 12 (1)*, 2001, pp. 60–67.

[4] F.Peper, J.Lee, J.Carmona, J.Cortadella, and K.Morita, "Brownian circuits: Fundamentals," in *ACM Journal of Emerging Technologies in Computing (JETC) (accepted)*, 2012.

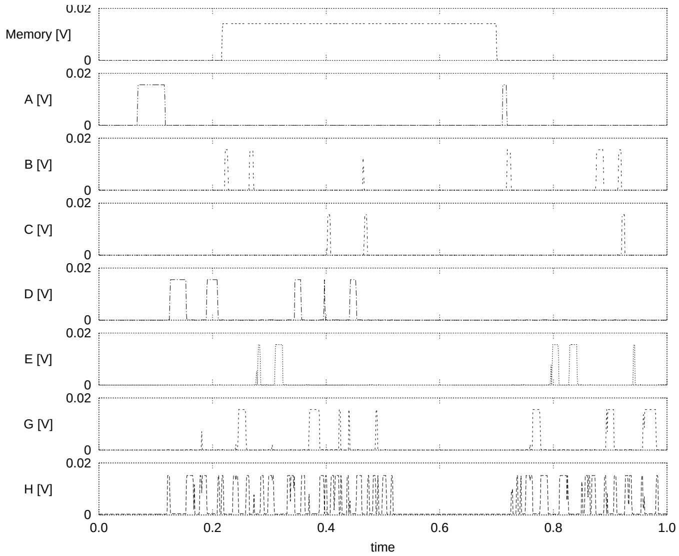

Fig. 14. Simulation results for Memory Enhanced Hub network example.

[5] J. Lee and F. Peper, "On brownian cellular automata," in *Proc. of Automata 2008, Luniver Press, UK*, 2008, pp. 278–291.

[6] J.Lee and F.Peper, "Efficient circuit construction in brownian cellular automata based on a new building-block for delay-insensitive circuits," in *Proc. 9th Int. Conf. on Cellular Automata for Research and Industry (ACRI10), S. Bandini, S. Manzoni, H. Umeo, G. Vizzari (Eds.), Lecture Notes in Computer Science 6350, Springer*, 2010, pp. 356–364.

[7] E.Bonabeau, G.Theraulaz, and M.Dorigo, "Swarm intelligence: From natural to artificial systems," in *Oxford University Press*, 1999.

[8] S. Safiruddin, S. D. Cotofana, F. Peper, and J. Lee, "Building blocks for fluctuation based calculation in single electron tunneling technology," in *Procedeengs of the 8th IEEE Conference on Nanothechnology*, August 2008.

[9] K. Likharev, "Single-electron devices and their applications," *Proceedings of the IEEE*, vol. 87, no. 4, pp. 606 –632, apr 1999.

[10] C. Wasshuber, "About Single-Electron Devices and Circuits," Ph.D. dissertation, TU Wien, 1998.

[11] S. Safiruddin and S. Cotofana, "Building Blocks for Delay-Insensitive Circuits using Single Electron Tunneling Devices," in *Proceeding of 7th IEEE International Conference on Nanotechnology (IEEE Nano)*, 2007, pp. 704–708.

[12] 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.

[13] C. Lageweg, S. Cotofana, and S. Vassiliadis, "Static Buffered set Based Logic Gates," in *Proceedings of the 2nd IEEE International Conference on Nanotechnology (IEEE Nano)*, Arlington, USA, 2002, pp. 491–494.

[14] ——, "A Linear Threshold Gate Implementation in Single Electron Technology," in *Proceedings of the IEEE Computer Society Workshop on VLSI*, Orlando, USA, 2001, pp. 93–98.

[15] 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.

[16] C. Meenderinck and S. Cotofana, "Computing Division Using Single-Electron Tunneling Technology," *IEEE Transactions on Nanotechnology*, vol. 6, no. 4, pp. 451–457, July 2007.

[17] C. Wasshuber, H. Kosina, and S. Selberherr, "SIMON - A Simulator for Single-Electron Tunnel Devices and Circuits," *IEEE Transactions on Computer-Aided Design*, vol. 16, no. 9, pp. 937–944, September 1997.