On the Design of SET Adders

Mawahib Sulieman and Valeriu Beiu

School of Electrical Engineering and Computer Science, Washington State University
102 Spokane Street (EME), Pullman, WA 99164-2752, USA
Email: {mawahib,vbeiu}@eecs.wsu.edu

ABSTRACT

Single-Electron-Technology (SET) is one of the future technologies distinguished by its small and low power devices. SET also provides simple and elegant solutions for threshold logic gates (TLG's). This paper presents the design of an optimal TLG adder implemented in SET. It provides a detailed procedure for designing capacitive–input SET TLG's for building the adder. The paper also presents design details and characteristics (delay and power dissipation) of a 16-bit Kogge-Stone SET adder.

Keywords: Single-electron technology, threshold logic, adders.

1 INTRODUCTION

Up to this date, integration and scaling provided lower costs and higher performance circuits. However, as devices become smaller, many physical effects retard the advance of micro/nanoelectronics towards higher performance systems. Owing to the difficulty of successfully scaling conventional bulk CMOS technology to meet the increased performance, density and reduced power dissipation required for future technology generations, new technologies are being researched. Single-Electron-Technology (SET) is one of the emerging technologies, and is distinguished by *a very small device size* and *ultra-low power dissipation*. These two properties promise to allow large-density integration without exceeding the power density physical limits [1]. Besides, SET provides simple and elegant solutions for implementing threshold logic gates (TLG's). A TLG is more powerful than a Boolean gate and its principle of operation is different [2]. TLG's implement threshold functions expressed as:

$$f\left(x_1,\ldots,x_\Delta\right) = sign\left(\sum\nolimits_{j=1}^{\Delta} w_j x_j + \theta\right) \qquad (1)$$

where w_j, θ, Δ are the weights, the threshold and the *fan-in* respectively. The TLG computes the weighted sum of its inputs, and compares this sum with a threshold value. If the sum is higher than the threshold, the TLG outputs a one, otherwise the output becomes a zero.

One of the SET logic circuits which has been the focus of several recent studies is the classical full adder [3]–[6]. The focus of these reports has been on a single-bit adder, and only a few articles extend it to multi-bit adders [3]. Even more, such extensions used simple adder structures and did not investigate advanced adder architectures.

In this paper we will describe an optimal structure for parallel-prefix TLG adders and its implementation in SET. At the gate level, the capacitive-input SET inverter (C-SET) lends itself well to the design of TLG's, and it has been used to design majority gates. This paper will generalize the design to arbitrary TLG's, and provide a detailed procedure for designing C-SET TLG's. With regard to characterizing the adder, we shall present and discuss both the delay and the power dissipation of our novel adder. In the literature about SET, power dissipation was only investigated for single gates (inverters), and was not reported for larger systems.

This paper is organized as follows. The architecture of the adder is described in section 2. This is followed by a detailed procedure for designing C-SET TLG's in section 3. Section 4 presents the simulation results of the 16-bit adder. Concluding remarks are provided in section 5.

2 ADDER ARCHITECTURE

Addition is among the functions which allow for simpler TLG implementations, *i.e.* shallow depth and polynomial size, while trading the size of weights for *fan-ins.* The sum function which is traditionally expressed as an XOR function, can also be represented in a linearly separable form. This can be done by writing it in terms of the *carry-in* (c_{i-1}) and *carry-out* (c_i) :

$$s_i = (a_i \cdot b_i \cdot c_{i-1}) + [c_i' \cdot (a_i + b_i + c_{i-1})] \qquad (2)$$

which may be represented as:

$$s_i = \text{sign}\ [a_i + b_i + c_{i-1} - 2c_i - 0.5] \qquad (3)$$

having weights (1,1,1,2). This solution was detailed in 1969 [7], used later in 1997 [8], and rediscovered in 1999 [9]. All the fastest TLG adder solutions use this (1,1,1,2) TLG to produce the sum. This alleviates the need for an XOR gate that requires two layers of TLG's. The fastest TLG adders also use fast parallel-prefix architectures. The main difference between these ultra fast adders is their way of implementing the *carry-merge* stages. Basically, two different approaches have been proposed for designing the *carry-merge* layers in fast TLG adders.

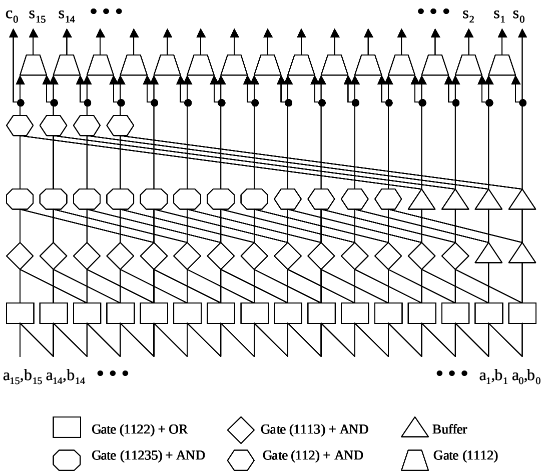

Fig. 1. TLG-optimized 16-bit adder.

One approach is based on *Fibonacci-weighted* TLG's: 1,1,2,3,… (*FIB*) [10], while the other uses *power-of-two* TLG's: 1,1,2,2,… (*PT*) [11]. The *FIB* approach has certain advantages. First of all, such TLGs can be immediately used in all the well-known architectures for Boolean adders. Secondly, its basic TLG has a small sum-of-weights for practical values of the *fan-in*. The main disadvantage is that it requires a first layer for computing the *propagate (p)* and *generate (g)* bits for each group of addend and augend bits. Obviously, this layer increases the overall depth of a *FIB* adder by one (when compared to *PT* adders). The *PT* approach does not require this first layer but its basic gate has a larger sum-of-weights than the *FIB* TLG. Our adder uses *PT* TLG's in the first layer and interfaces them to upper *FIB* layers being an optimal hybrid solution [12].

The adder architecture is shown in Fig. 1. It is based on the Kogge-Stone [13] adder. For $n = 16$, the adder consists of five layers:

- The first layer is a *PT* layer comprising the (1,1,2,2) TLG which produces 2-bit generate signals (G_i); and 2-input OR gates for the propagate signals (P_i).
- The second layer works as an interface to the upper *FIB* layers. The G_i TLG in this layer has *weights* (1,1,1,3). The P_i TLG's are simple majority gates implementing AND functions.
- The third and fourth layers are *FIB* layers and constitute the *carry-merge* tree. The third layer contains two types of TLG's:
  - A 3-input *FIB* TLG (1,1,2) performs a radix-2 carry merge $G_i = G_i + P_i \cdot G_{i-1}$.
  - A 5-input *FIB* TLG (1,1,2,3,5) performs a radix-3 carry merge $G_i = G_i + G_{i-1} \cdot P_i + G_{i-2} \cdot P_{i-1} \cdot P_i$. It also contains AND gates for the propagate signals (P_i).
- The fourth *FIB* layer contains only one type, namely the 3-input TLG (1,1,2).
- The last layer is the well-known TLG sum layer (1,1,1,2) described previously.

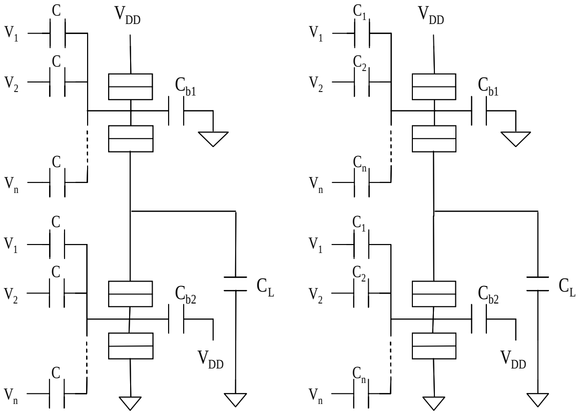

Fig. 2 Majority (left) and general TLG (right) capacitive-input SET gates.

3 SET THRESHOLD LOGIC GATES DESIGN

The TLG's design is based on the capacitive-input SET inverter. This structure was introduced for FET transistors in 1966 [14], and rediscovered in 1992 [15]. Since then, it has been known as the neuron-MOS (or νMOS). In [3] the application of this structure to majority SET gates was presented. In that article the adequacy of this approach to SET circuits was demonstrated and a full adder example was given.

The first step in our design was to augment the majority based design by generalizing it to arbitrary TLG's as shown in Fig. 2. With this modification, the new design reduces the number of components, the delay and power dissipation. Table 1 compares the characteristics of full adders based on majority gates and TLG's. The results were obtained using SIMON which is a Monte Carlo simulator for SET [16].

The TLG design is based on the SET inverter proposed by Tucker [17], and the inverter parameters are chosen such as to produce a step characteristic [3]. Since each TLG is based on an inverter, the output will always be the complement of the desired function. One approach to design a layered structure based on this TLG is to insert inverters between consecutive layers. A better solution is to implement the desired function in one layer and the dual of the function in the next layer. The first layer will produce the output complements, while the next layer will take the complements as inputs, implement the complement of the function and produce the desired function at the inverter output. This alternation continues for all remaining layers.

To design one TLG, we first determine the threshold equation of the particular function and/or the dual of the function. Secondly, the weight ratios are used to calculate the values of the capacitors. As an example we will show here the design of the TLG used in the last layer of the

	Majority	TLG
Number of components	57	28
Adder Delay (ns)	0.26	0.20
Power Consumption (pW)	1..95	0.75

Table 1: Comparison between Majority and TLG full adders.

adder (1,1,1,2). The threshold equation was given in equation (3). Eliminating the negative weight it can be written as: $s_i = a_i + b_i + c_{i-1} + 2c_i' - 2.5$. Hence, the dual of this function can be written as: $s_i' = a_i' + b_i' + c_{i-1}' + 2c_i - 2.5$. The fact that s_i and s_i' have equal weights and thresholds translates into identical TLG's.

The second step is to determine the capacitor values. These are calculated according to the weights and the sum of input capacitance. The sum is determined as part of the inverter parameters which produce a step inverter. Using the parameters given in [3], the sum is 3 aF. To calculate the capacitors, define C as the unit capacitance corresponding to a weight of 1. Since the threshold is 2.5, three capacitor units should be larger than 1.5 aF and two units less than this value. Mathematically: $3C > 1.5$ and $2C < 1.5$, hence $0.5 < C < 0.75$. For $C = 0.6$, the input capacitors are: $C_1 = C_2 = C_3 = 0.6$ aF and $C_4 = 1.2$ aF. Some TLG's require a bias capacitor in addition to the input capacitors. For example, consider the function implemented by the 3-input Fibonacci TLG (1,1,2). This function can be described as: $G_i = 2G_i + P_i + G_{i-1} - 1.5$ and its dual as: $G_i' = 2G_i' + P_i' + G_{i-1}' - 2.5$. In the adder described above, we implemented both G_i and G_i'. The latter with a threshold of 2.5 requires $0.5 < C < 0.75$. Using $C_1 = C_2 = 0.6$ and $C_3 = 1.2$ gives a total of 2.4 aF. Hence this TLG requires a bias capacitor of 0.6 aF which should be connected to ground for proper TLG operation. Table 2 shows the capacitor values used for all TLG's that constitute our 16-bit adder. Several AND/OR gates are used in this adder and each one has equal input capacitances and differs only in the values of bias capacitors. The parameters common to all gates are: $C_{b1} = C_{b2} = 9.0$ aF, $C_L = 24$ aF, $V_{dd} = 6.5$ mV.

TLG	C_1	C_2	C_3	C_4	C_5	C_b
(1,1,2,2)	0.4	0.4	0.8	0.8	-	0.6
(1,1,1,3)*	0.45	0.45	0.45	1.35	-	0.6
(1,1,2,3,5)	0.2	0.2	0.4	0.6	1.0	0.6
(1,1,2)	0.7	0.7	1.4	-	-	0.4
(1,1,2)*	0.6	0.6	1.2	-	-	0.6
(1,1,1,2)	0.6	0.6	0.6	1.2	-	-

Table 2: Capacitor values (in aF) for our 16-bit adder TLG's (* means the dual of the function).

4 ADDER CHRACTERISTICS

The 16-bit optimal adder was fully constructed, and was simulated using SIMON [16]. Due to the limited user interface, a MATLAB program was written to facilitate building the adder circuit. The program consists of two main modules, one to build the circuit from elementary gates and the other to specify stimuli signals. Simulation results showed that the adder functions properly.

Fig. 3 shows the delay of the 16-bit adder. The LSB of one input has a transition from '0' to '1' at 10 ns, and the output bits follow after some delay. Since the simulator is based on stochastic processes, the delay was calculated as the average of the delays for different random numbers. This 'average' delay is about 2 ns.

The power dissipation of the adder was thoroughly investigated. Fig. 4 shows the power of the 16-bit adder when running at different frequencies. The values obtained agree with reported results for SET inverters [18], taking into consideration the differences in load capacitance, voltage supply and scaling by the number of gates.

The simulation results mentioned above were obtained at helium temperature (0÷4 K). For getting an insight into the scaling of the power with respect to temperature, we have simulated one inverter at different temperatures. The results are shown in Fig. 5. The total power is obviously increased by temperature. This is due to the increase in static power caused by thermally generated tunneling.

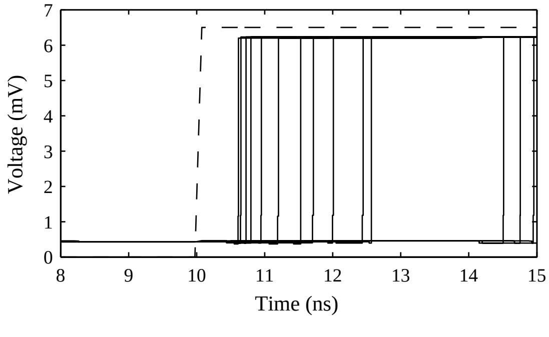

Fig. 3. Adder outputs (solid lines) and LSB of one input (dashed line)

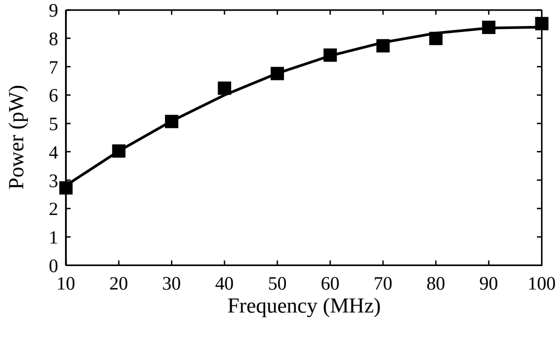

Fig.4. Adder power dissipation (with second order fitting).

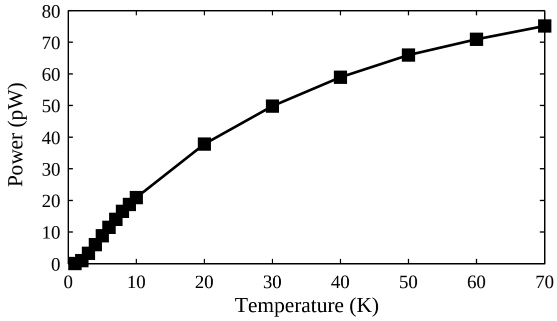

Fig. 5. Inverter power dissipation vs temperature.

With regard to integration, these results show that at liquid helium temperatures, an IC with 10^{11} transistors should dissipate below 1 W. As the temperature increases, a limit is reached where the gate does not function properly. Increasing the voltage supply can restore the functionality but will increase the power dissipation. Nevertheless, even if the supply voltage is increased 10 times (65 mV), an IC with 10^{10} devices should dissipate about 75 W.

5 CONCLUSION

A 16-bit adder was designed using TLGs. Each adder node consists of one or two capacitive-input SET TLG's. This can be compared to relatively complex Boolean gates used in *carry-merge* stages in CMOS adders. Simulation results showed quantitatively the ultra-low power dissipation of SET circuits. The circuit delay is high when compared to CMOS, as was expected for SET devices. The major problems encountered while doing this work were:

- The limited user interface of SIMON, which made it cumbersome to build the circuit.
- The very long simulation run which is typical of Monte Carlo based simulators. This could be alleviated by using SPICE simulations: either using specific models for SET [19], or by using a universal device model [20] on which a specific SET model can be defined.

REFERENCES

[1] V. Zhirnov, R. Cavin, J. Hutchby and G. Bourianoff, "Limits to binary logic switch scaling—A gedanken model," *Proc. IEEE*, vol. 91, Nov. 2003, pp. 1934–1939.

[2] S. Muroga, *Threshold Logic and Its Applications.* NewYork: John Wiley & Sons, 1971.

[3] H. Iwamura, M. Akazawa and Y. Amemiya, "Single-electron majority logic circuits," *IEICE Trans. Electron*., vol. E81-C, Jan. 1998, pp. 42–48

[4] Y. Ono, H. Inokawa and Y. Takahashi, "Binary adders of multi-gate single-electron transistor: Specific design using pass-transistor logic," *IEEE Trans. Nanotech.*, vol. 1, Jun. 2002, pp. 93–99.

[5] C. Lageweg, S. Coţofană and S. Vassiliadis, "A full adder implementation using SET based linear threshold gates," *Proc. Intl. Conf. Electronics, Circuits and Systems*, Sep. 2002, pp. 665–668.

[6] T. Oya, T. Asai, T. Fukui and Y. Amemiya, "A majority logic device using an irreversible single-electron box," *IEEE Trans. Nanotech.*, vol. 2, Mar. 2003, pp. 15–22.

[7] R. Betts, "Majority logic binary adder," U.S. Patent 3 440 413, Apr. 22, 1969.

[8] S. Coţofană and S. Vassisliadis, "Low weight and fan-in neural networks for basic arithmetic operations," *Proc. IMACS World Congress Sci. Comp., Modeling and Appl. Maths.*, vol. IV, 1997, pp. 227–232.

[9] J.F. Ramos and A.G. Bohórquez, "Two operand binary adders with threshold logic," *IEEE Trans. Comp.*, vol. 48, Dec. 1999, pp. 1324–1337.

[10] V. Beiu, "Neural addition and Fibonacci numbers," *Proc. Intl. Work-conf. Artif. Neural Networks,* Springer, LNCS 1607, vol. II, 1999, pp. 198–207.

[11] S. Vassiliadis, S. Coţofană, and K. Berteles, "2–1 addition and related arithmetic operations with threshold logic," *IEEE Trans. Comp*., vol. 45, Sep. 1996, pp. 1062–1067.

[12] M. Sulieman and V. Beiu, "Optimal practical adders using perceptrons," *Intl. Conf. Neural Networks and Signal Proc.* Nanjing, China, Dec. 2003, to appear.

[13] P.M. Kogge, and H.S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Trans. Comp*., vol. 22, 1973, pp. 783–791.

[14] J.R. Burns, "Threshold circuit utilizing field effect transistors," U.S. Patent 3 260 863, Jul. 12, 1966.

[15] T. Shibata and T. Ohmi, "Functional MOS transistor featuring gate-level weighted sum and threshold operation," *IEEE Trans. Electron Dev.*, vol. 39, Jun. 1992, pp. 1444–1455.

[16] C. Wasshuber, H. Kosina and S. Selberherr, "SIMON: A simulator for single-electron tunnel devices and circuits," *IEEE Trans. Comp. Aided Design of Integ. Circ. and Sys.*, vol. 16, Sep. 1997, pp. 937–944.

[17] J.R. Tucker, "Complementary digital logic based on the Coulomb blockade," *J. Appl. Phys*., vol. 72, Nov. 1992, pp. 4399–4413.

[18] Y-H Jeong, "Power consumption considerations of C-SET logics for digital application," *Proc. Intl. Conf. Solid-State and Integrated Circuits*, vol. 2, 2001, pp. 1373–1377.

[19] S.-H. Lee, "A practical SPICE model based on the physics and characteristics of realistic single-electron transistors," *IEEE Trans. Nanotech.*, vol. 1, Dec. 2002, pp. 226–232.

[20] M. Ziegler, G. Rose and M.R. Stan, "A universal device model for nanoelectronic circuit simulation," *Proc. IEEE Conf. Nanotech*, Aug. 2002, pp. 83–88.