A Fan-in Bounded Low Delay Adder for Nanotechnology

Yichun Sun and Meghanad Wagh

Department of ECE, Lehigh University, Bethlehem, PA 18015, yis205, mdw0@Lehigh.Edu

ABSTRACT

Adder is an important functional unit of a computer. This paper provides design of a low delay threshold adder (LDTA) using fan-in bounded gates in nanotechnology. An n-bit LDTA with a fan-in bound of $M+1$ has delay of $O(\log(N/M))$ and hardware complexity of $O(N\log(N/M))$. This design strategy allows a trade-off between the fan-in bound, delay and the complexity.

Keywords: adder, threshold gate, fan-in, low delay

1 INTRODUCTION

According to the *International Technology Road-map for Semiconductors* [1], [2], within another two decades, digital circuits based on nanoelectronic technologies of resonant tunneling diodes/transistors (RTD/RTT) [3], single electron transistor(SET) [4], [5] and quantum dot cellular automata (QCA) [6] will start replacing those based on the complementary metal oxide semiconductor (CMOS) technology. The nanoelectronic circuits will have higher speed, smaller size and lower power dissipation. While the elementary logic blocks in these technologies have already been fabricated and evaluated, the design techniques to create larger systems with these devices have not yet been fully developed.

The basic logic block implemented by all the three nanoelectronic technologies mentioned here is a threshold gate, which is much more powerful than the logic gates built in the CMOS technology [3]–[6]. Boolean gates AND/OR/NOT/NAND/NOR are threshold functions. Thus one can always express the conventional designs using threshold gates thereby providing their nanoelectronic implementation. However, this design translation fails to harness the true power of the threshold gate which allows one to realize very complex Boolean expressions, often through single gates.

Previous researchers have provided adder designs using RTD [7], SET [8] and QCA [9], [10] devices. But these and other prior works merely convert the designs developed for the CMOS technology into threshold logic suitable for the newer technologies. This paper develops strategies to design adder architectures directly with threshold gates. This allows one to exploit the full potential of threshold gates and improve certain required characteristics of the designs. Further, it is known that the reliability of gates based on nanotechnologies decreases as their fan-in (number of inputs to a gate) increases [11]–[13]. This work therefore assumes that the fan-in of the threshold gates is bounded. We compare the delay and complexity of our adder to those of the conventional adders (also implemented with threshold gates) and argue that only our adder allows a trade-off between these properties and reliability.

2 PRELIMINARIES

A Boolean function $f(x_1,\ldots,x_n)$, is called a *threshold function* if there exist real numbers w_1, w_2, …, w_n and T such that

$$f(x_1,x_2,\ldots,x_n)=\begin{cases}1 & \text{if } \sum_{i=1}^{n} w_i x_i \geq T\\ 0 & \text{otherwise.}\end{cases} \qquad (1)$$

The constants w_1, …, w_n and T are called the *weights* of the inputs and the *threshold* respectively. This function is denoted by $\mathbf{TH}(x_1,\ldots,x_n;w_1,\ldots,w_n;T)$. Since scaling of weights and threshold by the same amount has no effect of the inequality in (1), we use integer values for weights and the threshold. A *Majority function* is a threshold function with each $w_i=1$ and $T=\lfloor n/2\rfloor+1$. A majority function is denoted by $\mathbf{MAJ}(x_1,x_2,\cdots,x_n)$.

Two properties of the threshold functions used here are stated in the following theorem without proof.

Theorem 1 *P1.* $x_1x_2+x_3(x_1+x_2)=\boldsymbol{MAJ}(x_1,x_2,x_3)$. *P2. Let* $f=\boldsymbol{TH}(x_1,x_2,\ldots,x_n;w_1,w_2,\ldots,w_n;T)$ *with each* $w_i\geq 0$. *Then a function* g *of* $n+2$ *variables including two new variables* x_{n+1} *and* x_{n+2} *defined as* $g=x_{n+1}x_{n+2}+(x_{n+1}+x_{n+2})f$ *is also a threshold function. In particular,*
$g=\boldsymbol{TH}(x_1,\ldots,x_n,x_{n+1},x_{n+2};w_1,\ldots,w_n,w,w;T')$, *where the threshold* T' *of* g *is chosen to satisfy* $T'\geq 2T$ *and* $T'>\sum_{i=1}^{n} w_i$ *and* $w=T'-T$.

3 CARRY COMPUTATION WITH THRESHOLD FUNCTIONS

Let g_i and p_i denote the carry generation and propagation properties of the i-th bit position as in the Group Carry Lookahead Adders (GCLA). One has $g_i=a_ib_i$

and $p_i = a_i \oplus b_i$, where a_i and b_i are the i-th bits of the two operands. In order to avoid the ExOr which is not a threshold function, define $t_i = g_i + p_i = a_i + b_i$. Then $g_i t_i = g_i$ and $g_i + t_i = t_i$. Theorem 1(P1) now gives

$$\begin{aligned} c_i &= g_i + p_i c_{i-1} = g_i + t_i c_{i-1} \\ &= g_i t_i + (g_i + t_i) c_{i-1} = \mathbf{MAJ}(g_i, t_i, c_{i-1}). \end{aligned} \tag{2}$$

To relate carry c_i to some other previous c_j, we generalize g_i and t_i to multiple bits.Let $G_{i:j}$, $i \geq j$ denote the proposition that carry c_i is generated in the bit positions j through i. Similarly, let $P_{i:j}$ denote the proposition that this group of bits propagates a carry. As before, define $T_{i:j} = G_{i:j} + P_{i:j}$. Note that $G_{i:i} = g_i$ and $T_{i:i} = t_i$. Clearly, $G_{i:j} = G_{i:j} T_{i:j}$ and $T_{i:j} = G_{i:j} + T_{i:j}$. We then have

$$c_i = G_{i:j} + P_{i:j} c_{j-1} = \mathbf{MAJ}(G_{i:j}, T_{i:j}, c_{j-1}). \tag{3}$$

$G_{i:j}$ and $T_{i:j}$ can be expressed with Gs and Ts of smaller groups of bits. For any k, $i \geq k > j$,

$$\begin{aligned} G_{i:j} &= G_{i:k} + G_{k-1:j} P_{i:k} & (4) \\ &= \mathbf{MAJ}(G_{i:k}, T_{i:k}, G_{k-1:j}). & (5) \end{aligned}$$

$$T_{i:j} = G_{i:j} + P_{i:j} = \mathbf{MAJ}(G_{i:k}, T_{i:k}, T_{k-1:j}). \tag{6}$$

Quantities $G_{i:j}$ and $T_{i:j}$ for smaller sets of bits can be determined directly from the input bits as follows. Carry is generated in bit positions j through i only if it is generated in one of these bits and is propagated through the higher bits. Thus,

$$G_{i:j} = g_i + p_i(g_{i-1} + p_{i-1}(g_{i-2} + \cdots p_{j+1}(g_j) \cdots)). \tag{7}$$

Using the fact that $g_k + p_k X = g_k + t_k X$ for any k and X, one can also write (7) as:

$$\begin{aligned} G_{i:j} &= g_i + t_i(g_{i-1} + t_{i-1}(\cdots t_{j+1}(g_j) \cdots)) \\ &= a_i b_i + (a_i + b_i)(a_{i-1} b_{i-1} + (a_{i-1} + b_{i-1})(\cdots \\ &\qquad a_{j+1} b_{j+1} + (a_{j+1} + b_{j+1})(a_j b_j) \cdots)) \\ &= \mathbf{TH}(a_j, b_j, a_{j+1}, b_{j+1}, \ldots, a_i, b_i; \\ &\qquad 1, 1, 2, 2, \ldots, 2^{i-j}, 2^{i-j}; 2^{i-j+1}). \end{aligned} \tag{8}$$

The last step in (8) is obtained from Theorem 1(P2).

The expression for $T_{i:j}$ can be obtained similarly as

$$\begin{aligned} T_{i:j} &= \mathbf{TH}(a_j, b_j, a_{j+1}, b_{j+1}, \ldots, a_i, b_i; \\ &\qquad 1, 1, 2, 2, \ldots, 2^{i-j}, 2^{i-j}; 2^{i-j+1} - 1). \end{aligned} \tag{9}$$

To obtain the carries directly from the input bits and c_{-1}, use (3) and Theorem 1(P2) to get

$$\begin{aligned} c_i &= G_{i:0} + c_{-1} T_{i:0} \\ &= a_i b_i + (a_i + b_i)(a_{i-1} b_{i-1} + (a_{i-1} + b_{i-1})(\cdots \\ &\qquad a_1 b_1 + (a_1 + b_1)(a_0 b_0 + c_{-1}(a_0 + b_0)) \cdots)) \\ &= \mathbf{TH}(c_{-1}, a_0, b_0, a_1, b_1, \ldots, a_i, b_i; \\ &\qquad 1, 1, 1, 2, 2, \ldots, 2^i, 2^i; 2^{i+1}). \end{aligned} \tag{10}$$

Similarly, c_i and c_j, $j < i$ are related by

$$\begin{aligned} c_i &= \mathbf{TH}(c_j, a_{j+1}, b_{j+1}, a_{j+2}, b_{j+2}, \ldots, a_i, b_i; \\ &\qquad 1, 1, 1, 2, 2, \ldots, 2^i, 2^i; 2^{i+1}). \end{aligned} \tag{11}$$

We will often refer to $G_{i:j}$ and $T_{i:j}$ as the G and T functions over the (bit index) range $[i : j]$.

4 A LOW DELAY THRESHOLD ADDER (LDTA)

The strategy to design the N-bit adder with a small delay using threshold functions with fan-in bound of $M + 1$ can be described as follows. For convenience, assume $M = 2^m$ and $N = 2^n$.

Low Delay Threshold Adder (LDTA) Design.

1. Obtain c_i, $0 \leq i < M/2$ using (10).

2. Obtain G and T over the range $[jM/2 + k : jM/2]$, $0 \leq k < M/2$, $1 \leq j < 2N/M$ using (8) and (9).

3. For each i, $0 \leq i < n - m$, for $1 \leq j < N/(2^i M)$ and $2^i M/2 \leq k < 2^{i+1} M/2$, obtain G and T over the range $[j2^{i+m} + k : j2^{i+m}]$ from G and T over ranges $[j2^{i+m} + k : j2^{i+m} + 2^{i+m-1}]$ and $[j2^{i+m} + 2^{i+m-1} - 1 : j2^{i+m}]$, $2^{i+m-1} \leq k < 2^{i+m}$, $1 \leq j < 2^{n-m-i}$ using majority gates as in (5) and (6).

4. Obtain carries $c_{(M/2)2^i + k}$, $0 \leq k < (M/2)2^i$ using carry $c_{(M/2)2^i - 1}$ and the appropriate G and T as in (3), $0 \leq i \leq n - m$.

5. Obtain the sum bits from the carries using $s_i = \mathbf{TH}(a_i, b_i, c_{i-1}, c_i; 1, 1, 1, -2; 1)$.

Fig. 1 shows the carry computation (steps 1 through 4 of the algorithm) in an 8-bit LDTA using threshold gates with a fan-in bound of 5. The threshold gates in this architecture are defined (for $i = 1, 2, 3$) as:
$A_0 : \mathbf{TH}(a_{2i}, b_{2i}; 1, 1; 2)$,
$A_1 : \mathbf{TH}(a_{2i}, b_{2i}, a_{2i+1}, b_{2i+1}; 1, 1, 2, 2; 4)$,
$B_0 : \mathbf{TH}(a_{2i}, b_{2i}; 1, 1; 1)$,
$B_1 : \mathbf{TH}(a_{2i}, b_{2i}, a_{2i+1}, b_{2i+1}; 1, 1, 2, 2; 3)$,
$C_0 : \mathbf{TH}(a_0, b_0, c_{-1}; 1, 1, 1; 2)$ and
$C_1 : \mathbf{TH}(a_0, b_0, a_1, b_1, c_{-1}; 1, 1, 2, 2, 1; 4)$.

Note that functions C_0 and C_1 in Fig. 1 correspond to step 1 of the procedure explained above. Functions A_0 and A_1 compute the G over the ranges $[2i : 2i]$ and $[2i + 1 : 2i]$ respectively as in step 2 of the procedure. Functions B_0 and B_1 compute the T over the same ranges. The four majority gates on the left in the second row of Fig. 1 compute the G and T functions over larger ranges by combining G and T functions over smaller ranges as described in step 3 of the procedure. Finally, the remaining majority gates in the figure are used to compute the carries as in step 4 of the procedure.

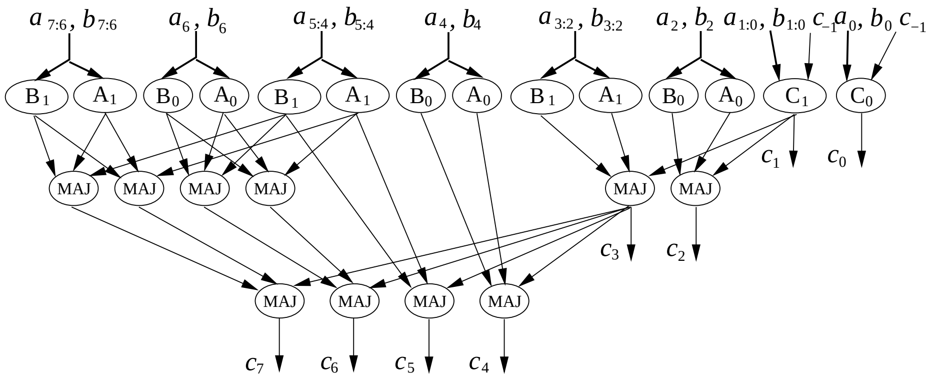

Figure 1: Computing all the carries of an 8-bit LDTA with a fan-in bound of 5.

Fig. 2 shows the RTD implementation of the carry computation in 8 bit LDTA with fan-in bound of 5.

Let $N = 2^n$ and $M = 2^m$. The complexity and delay of the LDTA is then given by the following Theorem.

Theorem 2 *The N-bit Low Delay Threshold Adder using threshold gates with fan-in bound $M + 1 \geq 5$ has a delay of* $3+\log(N/M)$ *and uses* $3N+N\log(N/M)$ *gates.*

Proof. Steps 1 and 2 of the LDTA design procedure follow directly from (8) - (10). The threshold functions used therein clearly satisfy the fan-in bound.

We now show that the G and T required in step 3 for any particular calculation for a given i, j and k are available either from step 2 or from a smaller i in step 3. For convenience, we denote the range $[j2^{i+m}+k : j2^{i+m}]$ in step 3, by $R(i, j, k)$. Similarly the range $[jM/2 + k : jM/2]$ used in step 2 is denoted by $R'(j, k)$. To compute G and T over $R(i, j, k)$, one needs T and G over two smaller ranges, namely, $[j2^{i+m} + k : j2^{i+m} + 2^{i+m-1}]$ and $[j2^{i+m} + 2^{i+m-1} - 1 : j2^{i+m}]$. One can verify that the second of these ranges is $R(i - 1, 2j, 2^{i+m+1})$. (For $i = 0$, this range is $R'(2j, 2^{i+m+1})$ in step 2.) Since Gs and Ts for $i - 1$ are computed before those for i, Gs and Ts over $R(i - 1, 2j, 2^{i+m+1})$ are available for the computation of Gs and Ts over $R(i, j, k)$.

To show that the Gs and Ts over the other smaller range, $[j2^{i+m}+k : j2^{i+m}+2^{i+m-1}]$, are also available for computing Gs and Ts over $R(i, j, k)$, we consider cases based upon the value of k. When $2^iM/2 \leq k < (2^i + 1)M/2$, Gs and Ts over the required range are available from step 2 when they are computed over the range $R'((2j + 1)2^i, k - 2^{i+m-1})$. When $i = 0$, this covers the entire k range. For other i values, when $(2^i + 2^t)M/2 \leq k < (2^i + 2^{t+1})M/2$, $0 \leq t < i$, the required range is $R(t, (2j + 1)2^{i-t-1}, k - 2^{i+m-1})$. Thus the required G and T over that range are available for computing Gs and Ts over $R(i, j, k)$.

Finally, carry computation in step 4 requires Gs and Ts over the range $[(M/2)2^i + k : (M/2)2^i]$. It is easy to show that when $i = 0$, these are computed in step 2 over the range $R'(1, k)$ and when $i > 0$, in step 3 over the range $R(i - 1, 1, k + 2^iM/2)$.

The complexity of LDTA can be obtained from the number of gates used in steps 1 through 5: $M/2$, $2N - M$, $N\log(N/M) - N + M$, $N - M/2$ and N. ∎

5 DISCUSSION AND CONCLUSION

This paper has developed a general strategy to design adder architectures directly in terms of threshold functions that are compatible with nanoelectronics. To increase the implementation reliability, we have assumed that the threshold gates used have a bounded fan-in. Our strategy is based on defining new logic primitives Gs and Ts of operand bits (Section 3) such that their evaluation as well as combination requires only threshold gates. These primitives may be computed over appropriate ranges of operand bits, either directly (see (8 and (9)) or by combining those over smaller ranges (see (4) - (6)). Once the appropriate Gs and Ts are available, the carries are computed from these with some previous carry (see (3)) or directly from the input operand bits and some previous carry (see (10) and (11)). Sum bits may be computed from the carries in one level of threshold gates. The chosen interdependence between carries (the carry chain) determines the delay and the hardware complexity of the adder. By experimenting with the carry chains, one can choose a suitable compromise between the complexity and the delay.

This paper has applied the strategy to obtain an adder architecture, LDTA, with a very low delay. This adder has 3-input majority gates on all but the first and the last levels. Table 1 compares the new LDTA architecture with CPA, CLA and GCLA.

One can see from the Table 1 that the LDTA is the only adder allowing a trade-off between the fan-in bound, delay and the complexity. It can give the minimum delay amongst all the adders. Through the fan-in bound, this adder allows the control of failure rates. Finally, it uses majority gates on all levels except the first

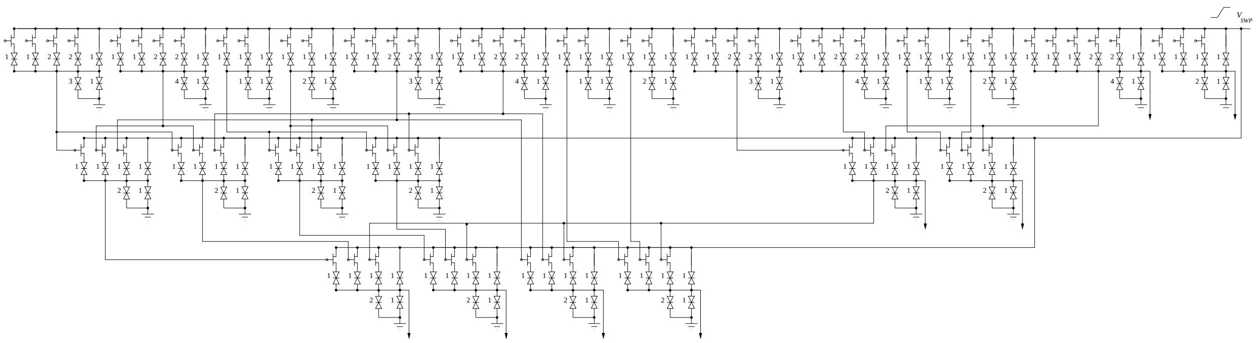

Figure 2: The 8 bit LDTA carry computation using Resonant Tunneling Diodes (RTD) with fan-in bound of 5.

Table 1. Comparison of N bit adders using nanotechnology devices with fan-in bound M.

Adder	Delay	Max fan-in	Number of gates
Carry Propagation Adder (CPA)	N	4	$2N$
Group Carry Lookahead Adder (GCLA)	$\log N + 2$	9	$(4/3)(N-1) + 3N$
Carry Lookahead Adder (CLA) [4]	$\log N$	$(N/4) + 1$	$(5/2)N$
new Low Delay Threshold Adder (LDTA)	$2 + \log(N/M)$	$2M + 1$	$O(N \log(N/M))$

(see Fig. 1), thus improving its manufacturability.

The strategy presented in this paper provides a new direction in adder architecture exploration. It is applicable to multiple nanotechnologies because it exploits the identical logic primitives in all these technologies. We believe that one can use the tools developed in this paper to design other new adders with the right balance of the delay and the complexity, all the while remaining within the realistic fan-in bounds.

REFERENCES

[1] J. A. Hutchby, G. I. Bourianoff, V. V. Zhirnov, and J. E. Brewer, “Extending the road beyond cmos,” *IEEE cir. and Dev. Mag.*, pp. 28–41, Mar. 2002.

[2] “The international technology roadmap for semiconductors: Emerging research devices.” http://www.itrs.net/, 2005.

[3] C. Pacha and K. Goser, “Design of arithmetic circuits using resonant tunneling diodes and threshold logic,” in *Proc. of the 2nd Workshop on Innovative Circuits and Systems for Nanoelectronics*, (Delft, NL), pp. 83–93, Sep. 1997.

[4] C. Lageweg, S. Cotofana, and S. Vassiliadis, “A linear threshold gate implementation in single electron technology,” in *Proc. IEEE-CS Annual Workshop on VLSI*, (Orlando, FL), pp. 93–98, Apr. 2001.

[5] A. Schmid and Y. Leblebici, “Robust circuit and system design methodologies for nanometer-scale devices and single-electron transistors,” *IEEE Trans. on VLSI Syst.*, vol. 12, pp. 1156–1166, Nov. 2004.

[6] I. Amlani, A. O. Orlov, G. Toth, G. H. Bernstein, C. S. Lent, and G. L. Snider, “Digital logic gate using quantum-dot cellular automata,” *Science*, vol. 284, pp. 289–291, April 1999.

[7] C. Pacha, U. Auer, C. Burwick, P. Glosekotter, A. Brennemann, W. Prost, F. J. Tegude, and K. F. Goser, “Threshold logic circuit design of parallel adders using resonant tunneling devices,” *IEEE Trans. on VLSI Syst.*, vol. 8, pp. 558–572, Oct 2000.

[8] C. R. Lageweg, *Single electron tunneling based arithmetic computation.* PhD thesis, Delft University of Technology, 2004.

[9] A. Vetteth, K. Walus, V. S. Dimitrov, and G. A. Jullien, “Quantum-dot cellular automata carry-look-ahead adder and barrel shifter,” in *IEEE Emerging Telecom. Tech. Conf.*, (Dallas), Sept. 2002.

[10] W. Wang, K. Walus, and G. A. Jullien, “Quantum-dot cellular automata adders,” in *3rd IEEE Conf. on NANO*, pp. 461–464, 2003.

[11] W. Prost, U. Auer, F.-J. Tegude, C. Pacha, K. F. Goser, G. Janssen, and T. van der Roer, “Manufacturability and robust design of nanoelectronic logic circuits based on resonant tunnelling diodes,” *Int. J. Circ. Theor. Appl.*, vol. 28, pp. 537–552, 2000.

[12] J. G. Guimaraes, H. C. Carmo, and J. C. da Costa, “Basic subcircuits with single-electon tunneling devices,” in *Proc. Symp. on Tech. and Devices*, 2002.

[13] C. Lageweg, S. Cotofana, and S. Vassiliadis, “Evaluation methodology for single electron encoded threshold logic gates,” in *Proc. Int. Conf. on Very Large Scale Syst. on Chip*, pp. 258–262, Dec. 2003.