

Logical Effort Delay Modeling of Sense Amplifier Based Charge Recycling Threshold Logic Gates

Peter Celinski^{1,2}, Sorin Cotofana² and Derek Abbott¹

¹The Department of Electrical and Electronic Engineering,
The University of Adelaide, SA 5005,
Australia.
celinski@eleceng.adelaide.edu.au

²Electrical Engineering Department,
Delft University of Technology, Mekelweg 4, 2628 CD Delft,
The Netherlands.

Abstract—In recent years, there has been renewed interest in Threshold Logic (TL), mainly as a result of the development of a number of successful implementations of TL gates in silicon, with improved performance and power dissipation compared to conventional logic. In this work, the problem of estimating the delay of circuits based on the Charge Recycling Threshold Logic (CRTL) gate implementation is addressed. A delay model is developed based on the recently proposed theory of Logical Effort. The model allows evaluation and comparison of high speed designs implemented in CRTL and conventional logic. The model is applied to the design of the 4-bit block carry-generate function and wide AND gates.

I. INTRODUCTION

Threshold logic (TL) was introduced over four decades ago, and over the years has promised much in terms of reduced logic depth and gate count compared to conventional logic-gate based design. Lack of efficient physical realizations has meant that TL has, over the years, had little impact on VLSI. Efficient TL gate realizations have recently become available, and a small number of applications based on TL gates have demonstrated its ability to achieve high operating speed and significantly reduced area [4], [6], [5], [2], [3], [10], [12].

The delay model developed in this work enables the prediction of the delay of CRTL based circuits and a systematic comparison of CRTL designs with conventional logic. Another motivator for developing the model is the desire to avoid the common and largely unsatisfactory presentation of circuit performance results commonly found in the literature in the form of delay numbers with insufficient information to allow comparison across different process technologies and loading conditions.

We begin in Section II by giving a brief overview of threshold logic. This is followed by a description of CRTL in Section III. Section IV reviews the theory of Logical Ef-

fort and the main result of the development of the proposed delay model is given in Section V, followed by example applications in VI. Finally a brief conclusion is given in Section VII.

II. THRESHOLD LOGIC

A threshold logic gate is functionally similar to a hard limiting neuron. The gate takes n binary inputs x_1, x_2, \dots, x_n and produces a single binary output y . A linear weighted sum of the binary inputs is computed followed by a thresholding operation.

The Boolean function computed by such a gate is called a threshold function and it is specified by the gate threshold T and the weights w_1, w_2, \dots, w_n , where w_i is the weight corresponding to the i^{th} input variable x_i . The binary output y is given by

$$y = \begin{cases} 1, & \text{if } \sum_{i=1}^n w_i x_i \geq T \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A TL gate can be programmed to realize many distinct Boolean functions by adjusting the threshold T . For example, an n -input TL gate with unit weights and $T = n$ will realize an n -input AND gate and by setting $T = n/2$, the gate computes a majority function. This versatility means that TL offers a significantly increased computational capability over conventional AND-OR-NOT logic. Significantly reduced area and increased circuit speed can therefore potentially be obtained, especially in applications requiring a large number of input variables, such as computer arithmetic. This is indicated by a number of practical results [9], [11], [6] which suggest advantages of TL over conventional Boolean logic.

III. CHARGE RECYCLING THRESHOLD LOGIC

The realization for CMOS threshold gates presented in [4] and used in the design of TL circuits in this work

is now described. Fig. 1 shows the circuit structure. The sense amplifier (cross coupled transistors M1-M4) generates output Q and its complement Q_b . Precharge and evaluate is specified by the enable clock signal E and its complement \bar{E} . The inputs x_i are capacitively coupled onto the floating gate ϕ of M5, and the threshold is set by the gate voltage t of M6. The potential ϕ is given by

$$\phi = \frac{\sum_{i=1}^n C_i x_i}{C_{tot}}, \quad (2)$$

where C_{tot} is the sum of all capacitances, including parasitics, at the floating gate of M5. Weight values are thus realized by setting capacitors C_i to appropriate values. Typically, in CMOS technology these capacitors are implemented between the polysilicon 1 and polysilicon 2 layers.

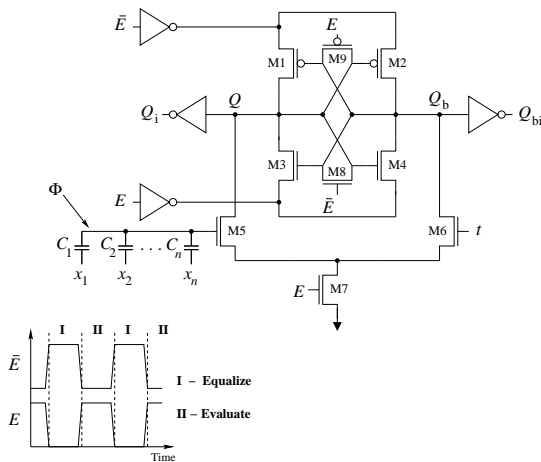


Fig. 1. The CRTL gate circuit and Enable signals.

The enable signal E controls the precharge and activation of the sense circuit. The gate has two phases of operation, the equalize phase and the evaluate phase. When \bar{E} is high the output voltages are equalized. When E is high, the outputs are disconnected and the differential circuit (M5-M7, M10, M11) draws different currents from the formerly equalized nodes Q and Q_b . The sense amplifier is activated after the delay of the enable inverters and amplifies the difference in potential now present between Q and Q_b , accelerating the transition. In this way the circuit structure determines whether the weighted sum of the inputs, ϕ , is greater or less than the threshold, t , and a TL gate is realized. Transistors M10 and M11 turn off the differential circuit after evaluation is completed to reduce the power dissipation. The gate was shown to reliably operate at high speed.

IV. LOGICAL EFFORT

Logical effort (LE) is a design methodology for estimating the delay of CMOS logic circuits, implementing a

given logic function [13], [14]. It provides a means to determine the best number of logic stages, including buffers, required to implement a given logic function, and to size the transistors to minimize the delay.

Logical effort is based on a reformulation of the conventional RC model of CMOS gate delay which separates the effects on delay of gate size, topology, parasitics and load. The relative simplicity of the method compared to other delay modeling techniques and sufficient accuracy allow it to be used early in the design process to evaluate alternative circuits.

The total delay of a gate, d , is comprised of two parts, an intrinsic parasitic delay p , and an effort delay, f , driving the capacitive load. The parasitic delay is largely independent of the transistor sizes in the gate, since wider transistors which provide increased current have correspondingly larger diffusion capacitances. The effort delay in turn depends on two factors, the ratio of the sizes of the transistors in the gate to the load capacitance and the complexity of the gate. The former term is called *electrical effort*, h , and the latter is called *logical effort*, g .

Electrical effort is defined as

$$h = \frac{C_{out}}{C_{in}}, \quad (3)$$

where C_{out} and C_{in} are the gate load capacitance and input capacitance, respectively. The logical effort, g , characterizes the gate complexity, and is defined as the ratio of the input capacitance of the gate to the input capacitance of an inverter that can produce equal output current. Alternatively, the logical effort describes how much larger than an inverter the transistors in the gate must be to be able to drive loads equally well as the inverter. By definition an inverter has a logical effort of 1.

The delay of a single logic gate can be expressed as

$$d = gh + p. \quad (4)$$

This delay is in units of τ , which is, the delay of an inverter driving an identical copy of itself, without parasitics. This normalization enables the comparison of delay across different technologies. The product gh is called the gate or stage effort.

The considerations so far apply to single gates, but may be extended to the treatment of delay through a path. Using uppercase to denote path parameters, the path electrical effort, H , is similarly defined as the ratio of the path load capacitance to the path input capacitance. The path logical effort, G , is given by

$$G = \prod g_i, \quad (5)$$

where the subscript i indexes the logic gates along the path. The effect of fanout, which causes some of the available drive current to be directed off the path being analyzed, is accounted for by considering the branching effort, b , which is defined as

$$b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}. \quad (6)$$

and the path branching effort is given by

$$B = \prod b_i. \quad (7)$$

Finally, the path effort, F , is given by the product of the path logical effort, the path branching effort and the path electrical effort

$$F = GBH. \quad (8)$$

The path delay, D , is the sum of the delays of each of the gate stages in the path, d_i , and consists of the *path effort delay*, D_F , and the *path parasitic delay*, P ,

$$\begin{aligned} D &= \sum d_i \\ &= D_F + P \\ &= \sum g_i h_i + \sum p_i. \end{aligned} \quad (9)$$

It can be shown that the path delay is minimized when each stage in the path bears the same stage effort and the minimum delay is achieved when the stage effort is

$$f_{min} = g_i h_i = F^{1/N}. \quad (10)$$

This leads to the main result of logical effort, which is the expression for minimum path delay

$$D_{min} = NF^{1/N} + P. \quad (11)$$

To equalize the effort borne by each stage in the path, the transistor sizes in each logic gate must be chosen according to the electrical effort given by Equation (10)

$$h_{i,min} = \frac{F^{1/N}}{g_i}. \quad (12)$$

This allows us to calculate the input capacitance and hence transistor size (width, assuming minimum length transistors) by applying the transformation

$$C_{in,i} = \frac{g_i C_{out,i}}{f_{min}}. \quad (13)$$

This input capacitance is distributed among the transistors within the gate connected to the input.

The preceding steps dictate how to size the gates along a path for minimum delay, taking into account the differing complexity of the gates as given by the logical effort.

Equally important is the selection of the correct number of stages. It has been shown [14] that for static CMOS logic, the near optimal stage effort is approximately 4, and stage efforts from 2.4 to 6 give delays within 15% of the minimum. Hence the best number of stages is approximately

$$N \approx \log_4 F. \quad (14)$$

For domino logic the optimal stage effort is 2 to 2.75 [14].

To minimize delay, the design should use the correct number of stages of logic and gates with low logical effort and parasitic delay. Path design may involve iteration, because the path's logical effort depends on the topology of individual gates, but the best number of stages is not known without knowing the path effort.

The simulated values of logical effort for a range of fanin NAND and NOR gates in a 0.18 μ m, 1.8V CMOS technology were shown to be significantly different from the theoretical value [16]. In the same work it was also shown that the delay value predicted by Equation (4) differed from simulation results on average by over 20% for the same range of gates, mainly as a result of the impact on delay time of the input transition times. However, the accuracy of the delay predicted by Equation (4) can be improved by calibrating the model by simulating the delay as a function of load (electrical effort) and fitting a straight line to extract τ , the inverter parasitic delay, p_{inv} , and the logical effort, g . We will use this technique to develop a calibrated logical effort based model for the delay of the CRTL gates.

V. MODELING CRTL DELAY

We begin by providing a set of assumptions which will simplify the analysis, a proposed expression for the worst case delay of the CRTL gate and a derivation of the model's parameters. The model is then applied to two practical circuit examples. The method described below may similarly be applied to other sense amplifier based linear threshold gates.

A. Notation and Assumptions

The TL gate is assumed to have n logic inputs (fanin), the total number of gate inputs connected to logic one is denoted by N , and T is the threshold of the gate. The potential of the gate of transistor M6, t , in Fig. 1 is given by

$$t = \frac{T}{n} \times V_{dd}. \quad (15)$$

In the worst case, the voltage ϕ in Equation (2) takes the values

$$\phi = t \pm \frac{\delta}{2} \quad (16)$$

where δ is given by

$$\delta = \frac{V_{dd}}{n}. \quad (17)$$

Equation (16) expresses the worst case (greatest delay) condition where the difference between ϕ and t is minimal, ie. the step voltage generated by the sum of inputs with respect to the threshold voltage is smallest. The value of $\phi = t - \delta/2$ corresponds to the rising and falling edges of the nodes Q and Q_b , respectively, in Fig. 1, and conversely for $\phi = t + \delta/2$.

The gate inputs are assumed to have unit weights, ie. $w_i = 1$, since the delay depends only on the value of N and T . Also, without loss of generality, we will assume positive weights and threshold, since negative weights may easily be accommodated in the differential structure of the gate by using a network of input capacitors connected to the gate of M6.

Since the gate is clocked, we will measure delay from the clock E to Q_i - Q_{bi} . Specifically, delay will be measured as the average of the 50% point on two falling transitions of E to the 50% points on the corresponding falling and rising edges of Q_i and Q_{bi} . Generally, the delay will depend on the threshold voltage, t , the step size, δ , and the capacitive output load on Q_i and Q_{bi} . To simplify the analysis, we will fix the value of t at 1.5 V. This value is close to the required gate threshold voltage in typical circuit applications. Therefore the worst case delay depends only on the fan-in and gate loading, and allows us to propose a model based on expressions similar to those for conventional logic based on the theory of logical effort.

B. Formulation of the Model and Parameter Extraction

The delay of the CRTL gate may be expressed as Equation (18). This delay is the total delay of the sense amplifier and the buffer inverters connected to Q and Q_b , and depends on the load, h , and the fanin, n , as follows

$$d_{E \rightarrow Q_i} = \{g(n)h + p(n)\}\tau. \quad (18)$$

The load, h , is defined as the ratio of load capacitance on Q_i (we assume the loads on Q_i and Q_{bi} are equal) and the CRTL gate unit weight capacitance. Both logical effort and parasitic delay in Equation (18) are a function of the fanin.

To determine the values of the parameters in Equation (18), we first determine the values of the parasitic delay of an inverter, p_{inv} . From Equation (4), the inverter delay is $d = \tau(gh + p_{inv})$, where by definition $g = 1$ for an inverter. To obtain the values of τ and p_{inv} , we may measure from HSPICE simulations the inverter delay for

TABLE I
DELAY PARAMETERS OF THE 0.35 μm , 3.3 V, 4M/2P
PROCESS AT 75°C.

τ	p_{inv}	FO4 delay
40 ps	1.18	204 ps

TABLE II
EXTRACTED CRTL GATE LOGICAL EFFORT, g , PARASITIC
DELAY, p , PARAMETERS FOR $n = 2$ TO 60 AND $h = 0$ TO 20
FOR THE 0.35 μm , 3.3 V, 4M/2P PROCESS AT 75°C AND
THE GATE DELAY NORMALIZED TO FO4 FOR $h = 1, 5$ AND
10.

n	g	p	$d_{E \rightarrow Q_i},$ $h=1$	$d_{E \rightarrow Q_i},$ $h=5$	$d_{E \rightarrow Q_i},$ $h=10$
2	0.346	2.5	0.55	0.82	1.15
5	0.357	3.3	0.71	0.98	1.33
10	0.365	4.0	0.84	1.13	1.48
15	0.376	4.3	0.90	1.19	1.56
20	0.375	4.7	0.98	1.27	1.63
30	0.400	5.0	1.04	1.35	1.74
40	0.424	5.1	1.07	1.40	1.80
50	0.439	5.2	1.09	1.43	1.85
60	0.460	5.2	1.09	1.45	1.90

various values of electrical effort h , and plot the delay versus h . The slope of this straight line gives the value of τ and the $h = 0$ axis intercept gives τp_{inv} .

The delay parameters for the industrial 0.35 μm process used to obtain the simulation results presented here and the simulated FO4 (fan-out of four) inverter delay are given in Table I. The value of τ is found to be 40 ps.

The values of g and p in Equation 18 were extracted by linear regression from simulation results for a range of fanin from $n = 2$ to 60 while the electrical effort was swept from $h = 0$ to 20 as shown in Table II. The Table also gives the absolute gate delay for three values of electrical effort, $h = 1, 5$ and 10, where h is the ratio of the load capacitance to the unit input capacitance of 3.37 fF.

By fitting a curve to the parameters g and p , CRTL gate delay may be approximated in closed form by

$$d_{E \rightarrow Q_i} = \{(0.002n + 0.34)h + \ln(n) + 1.6\}\tau. \quad (19)$$

It should be noted that the FO4 delay predicted by the LE model, $d = \tau(gh + P_{inv}) = 40(4 + 1.18) = 207\text{ps}$, agrees well with the simulated result of 204 ps. Additionally, the FO4 delay value is approximately 20% higher than that reported in the literature for a typical 0.35 μm process (see for example [8]), most likely due to the lower temperature used to obtain those results. For comparison,

the FO4 delay across various process technologies may be closely approximated by 500ps/micron (gate length) [8].

In order to use the parameters in Table II and Equation (19), it is necessary to compensate for the parasitic capacitance at the floating gate of M5. From Equation(2), the parasitic capacitance, C_p , contributes to a reduced voltage step, δ , on the gate of M5 in Fig. 1 with respect to the threshold voltage, t , as given by Equation (20),

$$\delta_{eff} = \left\{ \frac{\sum_{i=1}^n C_i}{\sum_{i=1}^n C_i + C_p} \right\} \delta_0, \quad (20)$$

where δ_0 is the nominal step given by Equation (17). This reduction in δ is equivalent to an increased value for the fanin. This effective fanin, n_{eff} , is given by

$$n_{eff} = \left\{ \frac{\sum_{i=1}^n C_i + C_p}{\sum_{i=1}^n C_i} \right\} n_0, \quad (21)$$

where n_0 is the number of inputs to the gate and n_{eff} is the value used to calculate the delay. Typically, for a large fanin CRTL gate, by far the major contribution to the parasitic capacitance will be from the bottom plate of the floating capacitors used to implement the weights. In the process used in this work, this corresponds to the poly1 plate capacitance to the underlying n-well used to reduce substrate noise coupling to the floating node.

For example, for a 32 input CRTL gate with 3.37 fF poly1-poly2 unit capacitors ($4\mu\text{m}^2$), the parasitic capacitance of poly1 to substrate is 29 fF, and the $\sum_{i=1}^n C_i = 32 \times 3.37 = 108$ fF. From Equation (21) the effective fanin to be used in the delay calculation is $((108+29)/108) \times 32 \approx 41$.

VI. APPLYING THE MODEL - DESIGN COMPARISON EXAMPLES

In order to illustrate the application of the model presented in the previous Section, the delay of a the 4-bit carry generate function used in adders, and wide AND gates used in ALUs designed using CRTL gates are evaluated and compared to the static and dynamic CMOS designs.

A. 4-bit Carry Generate

The carry generate signal, α , of a 4-bit block may be calculated using a single TL gate as follows [15]

$$\alpha = \text{sgn} \left\{ \sum_{i=0}^3 2^i (a_i + b_i) - 2^4 \right\}. \quad (22)$$

We assume a realistic load corresponding to $h=10$ (ie. $C_L=33.7$ fF). The sum of weights $N = 30$, so the worst case delay of this gate will correspond to the delay of a

gate of effective fanin, n_{eff} , of approximately 40. From Table II, the expected delay is 1.8 FO4, or 372 ps. Using Equation (19), the calculated delay is 379 ps.

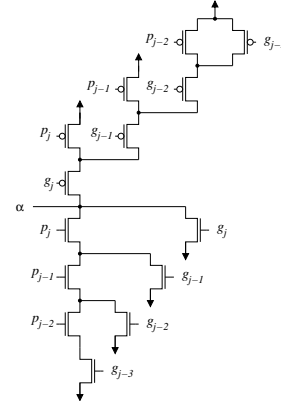


Fig. 2. The CRTL gate circuit and Enable signals.

The static CMOS gate used to compute the same function is shown in Fig. 2 [1]. To obtain a fair comparison, the transistors of the static CMOS gate were sized so that the CMOS gate input capacitance is equal to the input weight input capacitance of the CRTL gate, corresponding to $w_0=1$, or $C_{in}=3.37$ fF and the same load was used. The simulated slowest, g_{j-3} , input delay delay is 1.07 ns (5.3 FO4) and the fastest, g_j , input delay is 634 ps (3.1 FO4). The gate was then sized so that the input capacitance was equal the the largest input capacitance of the CRTL gate, corresponding to $w_3 = 8$, which is $8 \times 3.37=27$ fF. In this case the simulated slowest and fastest input delays are 514 ps (2.5 FO4) and 197 ps (0.96 FO4) respectively.

The dynamic CMOS implementation of the 4-bit carry generate function was also simulated, and the clock- α delay for the same load of 33.7 fF was measured, while maintaining an input capacitance equal to the unit weight CRTL gate input capacitance of 3.37 fF. The fastest input delay was 193 ps (0.94 FO4), while the slowest input was 460 ps (2.2 FO4). It should be noted that the above numbers exclude the delay of generating the p_i and g_i signals.

For the worst case slow-input delay, the CRTL has 18% lower delay than domino and 47% lower delay than static CMOS, for equal input capacitance and load.

B. Wide AND Gates

As a second example, we consider the design of wide AND gates. We will use the results for wide static CMOS AND gates presented in [14] as the basis for comparison.

Table III shows the FO4 delay for static CMOS AND gates with fanin from 8 to 64, for $H=1$ and $H=5$ [14], corresponding to the values of $h=1$ and $h=5$ in Table II.

Comparing Tables II and III, the CRTL gate design is on average 3 times faster and 2.8 times faster for $H=1$ and

TABLE III
 STATIC CMOS AND TREE DESIGNS AND FO4 DELAYS FOR
 MINIMUM DELAY FOR $n=8, 16, 32$ AND 64 FOR PATH
 ELECTRICAL EFFORT $H=1$ AND $H=5$

n	Tree	FO4 delay	
		H=1	H=5
8	4, 2	1.94	2.84
16	4, 4	2.58	3.38
32	4, 2, 2, 2	3.32	3.98
64	4, 2, 4, 2	3.86	4.54

$H=5$, respectively. Given that domino gates are typically 1.5 to 2 times faster than static gates [7], we could reasonably expect CRTL to be 1.5 to 2 times faster than the domino implementations. A detailed evaluation of domino wide AND tree delays is beyond the scope of this work.

VII. CONCLUSIONS

A delay model for Charge Recycling Threshold Logic gates based on the method of logical effort has been presented. The model was applied to two circuit design examples based on CRTL and conventional static and dynamic CMOS logic, and it was shown that the CRTL designs offer significantly reduced delay.

REFERENCES

- [1] A. Beaumont-Smith and C. Lim. Parallel prefix adder design. In *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, Vail, USA, June 2001.
- [2] P. Celinski, D. Abbott, and S. Al-Sarawi. Level sensitive latch. US Patent Number 6,542,016.
- [3] P. Celinski, S. D. Cotofana, and D. Abbott. A-DELTA: A 64-bit high speed, compact, hybrid dynamic-CMOS/Threshold-Logic adder. In *Proceedings of the 7th International Work Conference on Artificial and Natural Neural Networks (IWANN 2003)*, Lecture Notes in Computer Science, pages 73–80, Spain, June 2003.
- [4] P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott. Low power, high speed, charge recycling CMOS threshold logic gate. *IEEE Electronics Letters*, 37(17):1067–1069, August 2001.
- [5] P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott. Compact parallel (m,n) counters based on self timed threshold logic. *IEEE Electronics Letters*, 38(13):633–635, June 2002.
- [6] P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott. Low depth carry lookahead addition using charge recycling threshold logic. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 469–472, Phoenix, May 2002.
- [7] D. Harris and M. Horowitz. Skew-tolerant domino circuits. *IEEE Journal of Solid-State Circuits*, 32(11):1702–1711, November 1997.
- [8] M. Horowitz, C.-K. K. Yang, and S. Sidiropoulos. High-speed electrical signaling. *Micro, IEEE*, 18:12–24, 1998.
- [9] Y. Leblebici, H. Özdemir, A. Kepkep, and U. Çiliniroğlu. A compact high-speed (31-5) parallel counter circuit based on capacitive threshold-logic gates. *IEEE JSSC*, 31(8):1177–1183, August 1996.
- [10] H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, and U. Çiliniroğlu. A capacitive threshold-logic gate. *IEEE JSSC*, 31(8):1141–1149, August 1996.
- [11] M. Padure, S. Cotofana, and S. Vassiliadis. High-speed hybrid Threshold-Boolean logic counters and compressors. In *Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems*, pages 457–460, 2002.
- [12] M. Padure, S. Cotofana, and S. Vassiliadis. A low-power threshold logic family. In *Proc. IEEE International Conference on Electronics, Circuits and Systems*, pages 657–660, 2002.
- [13] I. Sutherland and B. Sproull. Logical effort: Designing for speed on the back of an envelope. In C. H. Sequin, editor, *Proceedings of the 1991 University of California Advanced Research in VLSI Conference*, pages 1–16. MIT Press, 1991.
- [14] I. E. Sutherland, R. F. Sproull, and D. L. Harris. *Logical Effort, Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.
- [15] S. Vassiliadis, S. Cotofana, and K. Bertels. 2-1 addition and related arithmetic operations with threshold logic. *IEEE Trans. Computers*, 45(9):1062–1067, September 1996.
- [16] X. Y. Yu, V. G. Oklobdzija, and W. W. Walker. Application of logical effort on design of arithmetic blocks. In *Proceedings of 35th Annual Asilomar Conference on Signals, Systems and Computers*, pages 872–874, November 2001.