Delay Evaluation of High Speed Data-Path Circuits Based on Threshold Logic

Peter Celinski[1], Derek Abbott[1], and Sorin D. Cotofana[2]

[1] Centre for High Performance Integrated Technologies & Systems (CHiPTec)
The Department of Electrical and Electronic Engineering,
The University of Adelaide, SA 5005, Australia.
celinski@eleceng.adelaide.edu.au
[2] Computer Engineering Group, Electrical Engineering Department,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands.
sorin@ce.et.tudelft.nl

Abstract. The main result is the development, and delay comparison based on Logical Effort, of a number of high speed circuits for common arithmetic and related operations using threshold logic. The designs include 8 to 64-input AND, 4-bit carry generate, and the carry-out of a (7,3) parallel (population) counter. The circuits are designed using both domino gates and the recently proposed CMOS Charge Recycling Threshold Logic (CRTL). It is shown that compared to domino, the CRTL design examples are typically between 1.3 and 2.7 times faster over a wide range of load values, while presenting the same input capacitance to the driver.

1 Introduction

As the demand for higher performance very large scale integration processors with increased sophistication grows, continuing research is focused on improving the performance and power dissipation of the arithmetic and other units.

The aim of this paper is firstly to propose a Logical Effort (LE) [3] based delay model for CRTL gates. Secondly, this model is used as the basis for demonstrating the performance advantage of Charge Recycling Threshold Logic [2] over conventional CMOS dynamic logic for a range of circuits used in processor datapaths. The proposed circuits include very wide AND, 4-bit carry, and (7,3) counter critical path. The main motivation for using an LE based delay comparison 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 processes and loading conditions.

We begin in Section 2 by giving a brief overview of threshold logic, including a description of CRTL. Section 3 briefly reviews Logical Effort and presents the delay model for CRTL gates. The circuit design examples are presented and evaluated in Section 4. Finally a conclusion and suggestions for future work are given in Section 6.

E. Macii et al. (Eds.): PATMOS 2004, LNCS 3254, pp. 899–906, 2004.

2 Threshold Logic AND CRTL

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. Efficient CMOS TL gate implementations have recently become available, and a small number of applications based on TL gates have demonstrated its ability to achieve high operating speed, low power dissipation and significantly reduced area [1].

A threshold logic gate is functionally similar to a hard limiting neuron. The gate takes n binary inputs x_1,x_2,...,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 specified by the gate threshold, T, and the weights w_1,w_2,...,w_n, where w_i is the weight corresponding to the i^{th} input variable x_i. The binary output y is given by

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

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 $T = n$ 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 NAND-NOR-NOT logic.

We now briefly describe the realization for CMOS threshold gates presented in [2]. Fig. 1 shows the circuit structure. The sense amplifier (cross coupled transistors M1-M4) generates output y and its complement y_i. Precharge and evaluate is specified by the enable clock signal E and its complement E_i. 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 $\phi = \sum_{i=1}^{n} C_i x_i / C_{tot}$, where C_{tot} is the sum of all capacitances, including parasitics, at the floating node. Weight values are thus realized by setting capacitors C_i to appropriate values. These capacitors may be implemented between the polysilicon-1 and polysilicon-2 layers.

The enable signal, E, controls the precharge and activation of the sense circuit. The gate has two phases of operation, the evaluate phase and the equalize phase. When E_i is high the output voltages are equalized. When E is high, the outputs are disconnected and the differential circuit (M5-M7) draws different currents from the formerly equalized nodes y and y_i. The sense amplifier is activated after the delay of the enable inverters and amplifies the difference in potential now present between y and y_i, accelerating the transition to full swing. 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. For full details, please refer to [2].

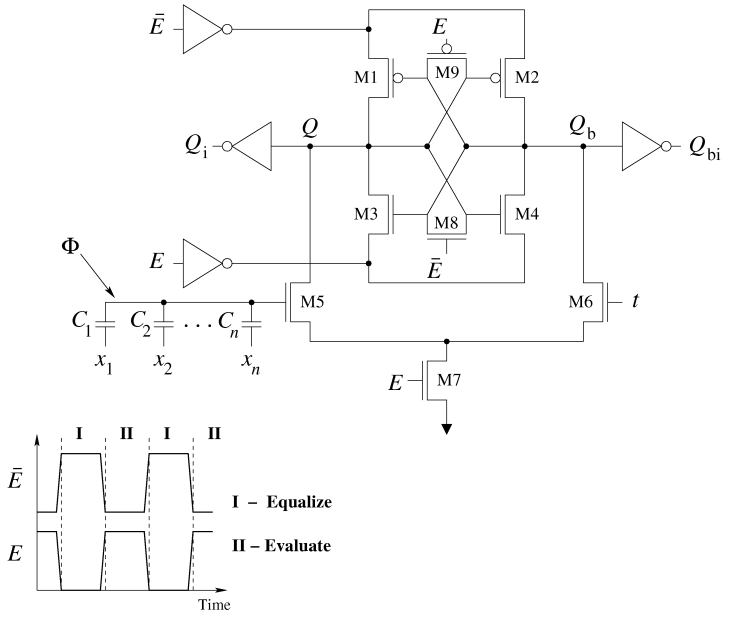

Fig. 1. The CRTL gate circuit and Enable signals.

3 Logical Effort

Logical effort (LE) is a design methodology for estimating the delay of CMOS logic circuits [3]. 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.

The total delay of a gate, d, is comprised of two parts, an intrinsic parasitic delay p, and an effort delay, $f = gh$:

$$d = (gh + p)\tau. \tag{2}$$

The delay unit τ 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 parasitic delay is largely independent of the transistor sizes in the gate. The effort delay, gh, depends on 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. 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. By definition an inverter has a logical effort of 1. These considerations may be extended to the treatment of delay through a chain of N gates, to minimize the total path delay for a given load and input capacitance.

The path effort, F, is given by the product of the path logical effort, the path branching effort and the path electrical effort, $F = GBH$. The path delay, D, is minimized when each stage in the path bears the same stage effort and the minimum delay is achieved when the stage effort is $f_{min} = g_i h_i = F^{1/N}$. This leads to the main result of logical effort, which is the expression for minimum path delay $D_{min} = NF^{1/N} + \sum p_i$.

The accuracy of the delay predicted by LE for any gate 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 parasitic delay, p, 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. For a full understanding of this paper the reader should be familiar with logical effort, for details refer to [3].

4 The CRTL Delay Model

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 TL gate is assumed to have n logic inputs (fanin) and T is the threshold of the gate. The potential of the gate of transistor M6, t, in Fig. 1 is given by $t = (T/n) \times V_{dd}$. In the worst case, the voltage ϕ takes the values $\phi = t \pm \delta/2$, where δ is given by $\delta = V_{dd}/n$.

The worst case (greatest delay) condition occurs when the difference between ϕ and t is minimal. 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 ϕ and t. Also, without loss of generality, we will assume positive weights and threshold. Since the gate is clocked, we will measure average 50% transition delay from the clock E to 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. The delay is not strongly dependent on the actual gate threshold, so the subsequent results are valid over a wide range of threshold voltages, t. Therefore, for modeling purposes, the worst case delay is assumed to depend 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.

The delay of the CRTL gate may be expressed as Equation (3). 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

$$d_{E \to Qi} = \{g(n)h + p(n)\}\tau. \tag{3}$$

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 (3) are a function of the fanin.

Table 1. 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	$d_{E\to Qi}$, h=1	$d_{E\to Qi}$, h=5	$d_{E\to Qi}$, 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

The gates used in this work are modeled using an industrial 0.35 μm, 3.3V, 4M/2P process technology at 75°C. The delay unit τ is 40 ps, p_{inv} is 1.18 and simulated FO4 (fan out of four inverter) delay is 204 ps.

The values of g and p in Equation (3) 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 1. 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.

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

$$d_{E\to Qi} = \{(0.002n + 0.34)h + \ln(\mathrm{n}) + 1.6\}\tau. \tag{4}$$

In order to use the parameters in Table 1 and Equation (4), it is necessary to compensate for the parasitic capacitance at the floating gate of M5. 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. This reduction in δ is equivalent to an increased value for the fanin. This effective fanin, n_{eff}, is given by

$$n_{eff} = \left\{ \frac{\sum_{i=1}^{n} C_i + C_p}{\sum_{i=1}^{n} C_i} \right\} n_0, \tag{5}$$

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 a 32-input CRTL gate with 3.37 fF poly1-poly2 unit capacitors ($4\mu\mathrm{m}^2$), the extracted layout 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 (5) the effective fanin to be used in the delay calculation is $((108+29)/108) \times 32 \approx 41$.

5 Design Examples and Comparison

In order to illustrate the application of the model presented in the previous Section, the delay of wide AND gates used in ALUs, the 4-bit carry generate function used in adders, and and the carry-out of a (7,3) parallel counter, designed using both domino and CRTL are evaluated and compared. In all examples the transistors of the domino circuits (first stage only for multi stage circuits) are sized to present the same input capacitance as the minimum sized inverter (1.8 μm of gate width, minimum length), which is approximately equal to the CRTL unit weight input capacitance, to ensure all designs affect the delay of the driver equally.

5.1 Wide AND

As the first example, we consider the design of wide AND gates used for example in ALUs for zero detection. The minimum delay domino trees for 8 to 64 inpus are listed in Table 2, e.g. 4, 4, 2 denotes a 3 layer tree design of the 32-input AND consisting of four 4-input AND gates in the input layer, four 4-input gates in the second layer and a 2-input gate in the third layer. These minimum delay trees were obtained by extracting the logical effort and parasitic delay of 2-, 4- and 8-input domino AND gates from simulations (see Chap. 5 of [3] for details), and finding the tree which minimizes the sum of effort and parasitic delay. The calibrated electrical effort and parasitic delay were used in the domino delay calculation.

Table 2 shows the FO4 delay for domino and CRTL AND gates with fanin from 8 to 64, for path electrical effort H=1 and H=10, corresponding to the values of h=1 and h=10 in Table 1. Note that increased vales of n_{eff} are used to obtain the correct CRTL delay values from Table 1.

Comparing Tables 1 and 2, the CRTL gate design is on average approximately 1.8 to 2.7 times faster than domino-CMOS for a path electrical effort of 10 and 1, respectively.

Table 2. Minimum delay domino-CMOS AND tree designs with fanin n=8, 16, 32 and FO4 delay comparison with CRTL for path electrical effort H=1 and 10

n	Domino tree	H=1		H=10	
		Domino	CRTL	Domino	CRTL
8	4, 2	1.91	0.84	2.34	1.48
16	4, 4	2.33	0.98	2.8	1.63
32	4, 4, 2	3.22	1.07	3.47	1.80
64	4, 4, 4	3.5	1.1	3.93	1.95
Average		2.74	1.0	3.14	1.72

5.2 4-Bit Carry Generate

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

$$\alpha = \text{sgn}\left\{\sum_{i=0}^{3} 2^i(a_i + b_i) - 2^4\right\}. \tag{6}$$

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. For H=10, corresponding to a 33.7 fF load capacitance, Table 1 gives the expected delay of 1.8 FO4, or 372 ps. Using Equation (4), the calculated delay is 379 ps.

The domino gate used to compute the same function for comparison is the well known Manchester-carry circuit. The electrical effort and parasitic delay for the slowest input, g_{j-3}, were extracted from simulation [3] and used to calculate the worst case delay for h=1 and h=10. The results are shown in Table 3. It should be noted that the domino gate delay numbers exclude the delay of generating the bitwise p_i and g_i signals.

Under the conditions of equal input capacitance and load, the CRTL gate is 1.3 to 1.6 times faster. This is a significant delay improvement even in this case of a function with a small number of logic inputs.

Table 3. 4-bit carry generate, G_j^{j-3}, and (7,3) counter c_{out} FO4 delay comparison with CRTL for path H=1 and 10

	H=1		H=10	
Function	Domino	CRTL	Domino	CRTL
G_j^{j-3}	1.7	1.07	2.42	1.8
(7,3) c_{out}	1.5	0.84	1.9	1.48

5.3 (7,3) Counter Critical Path

As the final design example, we consider the critical path of a (7,3) parallel counter, commonly used in multipliers. The domino critical path for c_{out} consists of two full adders. The CRTL implementation computes the majority function using a single gate, where the output is logic 1 if 4 or more inputs are 1. The delay results are shown in Table 3. The CRTL implementation is between 1.3 and 1.8 times faster.

6 Conclusions

A logical effort based delay model for CRTL gates was introduced and applied to the evaluation of a number of common datapath circuit elements. It was shown that compared to domino, the CRTL design examples are between 1.3 and 2.7

times faster over a wide range of loads, while presenting a significantly reduced input capacitance to the driver from the previous stage. The design examples used in the comparison were chosen to illustrate the typical performance gains of CRTL over domino which may be expected and this will be strongly application dependent. The important consideration of relative power dissipation is the subject of ongoing work.

References

1. P. Celinski, S. Cotofana, J. F.López, S. Al-Sarawi, and D. Abbott. State-of-the-art in CMOS threshold logic VLSI gate implementations and applications. In *Proceedings of the VLSI Circuits and Systems Conference*, volume 5117, pages 53–64, Spain, 2003. SPIE.
2. P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott. Low power, high speed, charge recycling CMOS threshold logic gate. *IEE Electronics Letters*, 37(17):1067–1069, August 2001.
3. I. E. Sutherland, R. F. Sproull, and D. L. Harris. *Logical Effort, Designing Fast CMOS Circuits*. Morgan Kaufmann, 1999.
4. 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.