Invited Paper

State-of-the-Art in CMOS Threshold-Logic VLSI Gate Implementations and Applications

Peter Celinski[a,c], Sorin D. Cotofana[c], José F. López[d], Said Al-Sarawi[a] and Derek Abbott[b]

[a] Centre for High Performance Integrated Technologies & Systems (CHiPTec)
[b] Centre for Biomedical Engineering (CBME)
Department of Electrical & Electronic Engineering
The University of Adelaide, SA 5005,
Australia.

[c] Computer Engineering Group
Electrical Engineering Department
Delft University of Technology
Mekelweg 4, 2628 CD Delft
The Netherlands.

[d] Research Institute for Applied Microelectronics
Universidad de Las Palmas de G.C.
35017-Spain. *

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 CMOS. This paper presents a summary of the recent developments in TL circuit design. High-performance TL gate circuit implementations are compared, and a number of their applications in computer arithmetic operations are reviewed. It is shown that the application of TL in computer arithmetic circuit design can yield designs with significantly reduced transistor count and area while at the same time reducing circuit delay and power dissipation when compared to conventional CMOS logic.

Keywords: Threshold logic, VLSI, computer arithmetic

1. INTRODUCTION

Threshold logic 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 Boolean logic-gate based design. However, 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,[1–7] and a number of applications based on TL gates have demonstrated its ability to achieve high operating speed and significantly reduced area.[4,6–8]

The focus of this work is a review of the significant recent developments in TL, including gate implementations and applications in computer arithmetic. The gates considered are compatible with conventional CMOS logic, and have been shown to be suitable for high performance digital applications. We begin in Section 2 by giving a brief overview of threshold logic. This is followed by a description, performance evaluation and comparison of recently proposed high-speed TL gate designs in Section 3. Section 4 surveys a number of recently reported applications of these gates in computer arithmetic. Finally, a conclusion is given in Section 5.

P. Celinski, corresponding author: celinski@eleceng.adelaide.edu.au, S. D. Cotofana: sorin@ce.et.tudelft.nl, José F. López: lopez@iuma.ulpgc.es, Said Al-Sarawi: alsarawi@eleceng.adelaide.edu.au, D. Abbott: dabbott@eleceng.adelaide.edu.au

2. OVERVIEW OF THRESHOLD LOGIC

Threshold logic emerged in the early 1960's as a generalized theory of switching logic and includes conventional Boolean logic as its subset. A threshold logic gate is functionally similar to a hard limiting neuron without learning capability. The gate takes n binary inputs $x_1,x_2,\ldots,x_n$ and produces a single binary output Y, as shown in Fig. 1.

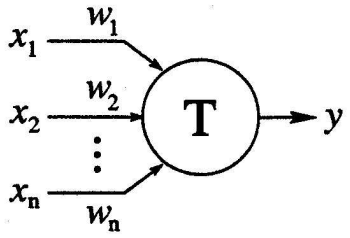

Figure 1. Threshold Gate Model

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 $w_1,w_2,\ldots,w_n$, where w_i is the weight associated with the i^{th} input variable x_i. The output y is given by (all operators algebraic):

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

This function can be written in a more compact form using the sgn notation:

$$y = \text{sgn}\left\{\sum_{i=1}^{n} w_i x_i - T\right\}, \tag{2}$$

where the sgn function is defined as follows, $\text{sgn}(x) = 1$ if $x \geq 0$ and $\text{sgn}(x) = 0$ if $x < 0$.

A device which implements this theoretical model must compute the linear weighted sum of the binary inputs, store the threshold value and compare the weighted sum to this threshold. The gates discussed here differ follow this paradigm, but they differ in the way they implement the weights, threshold and comparison. A TL gate can be programmed to realize many distinct Boolean functions by adjusting the threshold T and/or the weights w_i. 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 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 illustrated by a number of practical results[9–11] which suggest advantages of TL over conventional Boolean logic.

3. HIGH SPEED THRESHOLD LOGIC GATE IMPLEMENTATIONS

A number of recently proposed CMOS TL gate designs are now reviewed. Rather than presenting an exhaustive review of all previously reported gate designs of note, the main focus is on those designs which have led to the design of high-speed TL based circuits. The reason for this is to establish a framework for developing digital systems, based on CMOS TL gates, which have performance competitive with that of conventional static or dynamic-CMOS.

The requirements for high speed and high fan-in lead us to consider predominantly dynamic, differential gates. The gates we consider in our review may be classified as either voltage mode or current/conductance mode. The voltage mode designs discussed here include neuron-MOS (ν**MOS**)[12] (the only static gate considered), Capacitive Threshold Logic (**CTL**),[4] Latched neuron-MOS (**L-**ν**MOS**)[13] and Charge Recycling Threshold

Logic (**CRTL**).[6] The current mode designs include Latched Comparator Threshold Logic (**LCTL**),[3] Equalized Current-Mode Threshold Logic (**ECMTL**)[5] and Differential Current-Switch Threshold Logic (**DCSTL**).[7]

The important gate features are speed of operation, maximum sum-of-weights (or, equivalently, identically weighted fan-in), area and power dissipation. The maximum sum-of-weights sets the minimum signal level (voltage or current) which is required to be resolved by the comparator in the TL gate. A high value of fan-in generally leads to shallow depth logic networks and therefore reduced circuit delay. Another desirable feature is the ability to easily implement negative weights, since some arithmetic operations are conveniently expressed in the form of Equation 2 where the values of w_i are negative. Ease of dynamic re-programmability of the weights and threshold may also be important in reconfigurable logic operations.

3.1. Voltage Mode Implementations

The voltage mode TL gate implementations described here are based on the principle of the capacitive synapse proposed in.[14] The underlying concept is the use of an array of capacitors to implement the weighted sum of inputs, connected to the gate of a MOS transistor (the νMOS transistor[1] also known as the Multiple Input Floating Gate transistor). Typically, in CMOS technology these capacitors are implemented between the polysilicon 1 and polysilicon 2 layers. Figure 2 depicts the four voltage mode gates considered here, including νMOS, Capacitive Threshold Logic, Latched νMOS and Charge Recycling Threshold Logic.

3.1.1. νMOS

The νMOS transistor based static TL gate proposed in the early 1990s uses an array of capacitors to implement the input weights, followed by one or more inverters to implement the thresholding operation, as shown in Fig. 2(a).

The input of the first inverter in the chain in Fig. 2(a) is effectively floating, and its voltage is given by

$$\phi = \frac{\sum_{i=1}^{n} C_i x_i}{C_{tot}}, \tag{3}$$

where C_{tot} is the sum of all capacitances at the floating node, including parasitic capacitances. The switching point of the first inverter in the chain, the *primary inverter*, determines the gate threshold, and the subsequent inverters serve to generate a full swing rail-to-rail output voltage. This expression assumes that no charge is initially present on the floating node.

The presence of this charge is, however, unavoidable as a result of fabrication, and for this reason techniques such as UV erasure must be used.[1] The νMOS gate suffers from limited fan-in (typically <12) due to variability of the switching threshold of the primary inverter as a result of process variations. It is also relatively slow and has a high data dependent power dissipation as a result of the static current flowing from V_{dd} to GND in the primary inverter caused by the floating gate voltage. The gate is highly compact and relatively simple to design, and the gate threshold may be programmed by adding control capacitors.[12] Negative weights can not be implemented in the νMOS gate.

3.1.2. Capacitive Threshold Logic

To overcome the limited fan-in of νMOS, Capacitive Threshold Logic was proposed.[4] The circuit schematic of the CTL gate is shown in Fig. 2(b). An n-input CTL gate comprises n weight-implementing capacitors (C_i) followed by one or more inverters which function as voltage comparators to generate the binary output. The main difference between CTL and neuron-MOS lies in the way the value of the gate threshold is set. In CTL, the threshold value is a function of an external reference voltage V_{ref}.

The CTL gate operates in a two-phase non-overlapping clock scheme consisting of a reset phase ϕ_R and an evaluate phase ϕ_E. During the reset phase the row voltage ϕ is reset to the threshold voltage V_{th} of the primary inverter, while the capacitor bottom plates are set to the reference voltage V_{ref}. During the evaluation phase, the row voltage is perturbed from V_{th} by the inputs x_i which now become capacitively coupled onto the effectively floating input to the primary inverter. The magnitude of this perturbation is a function of V_{ref},

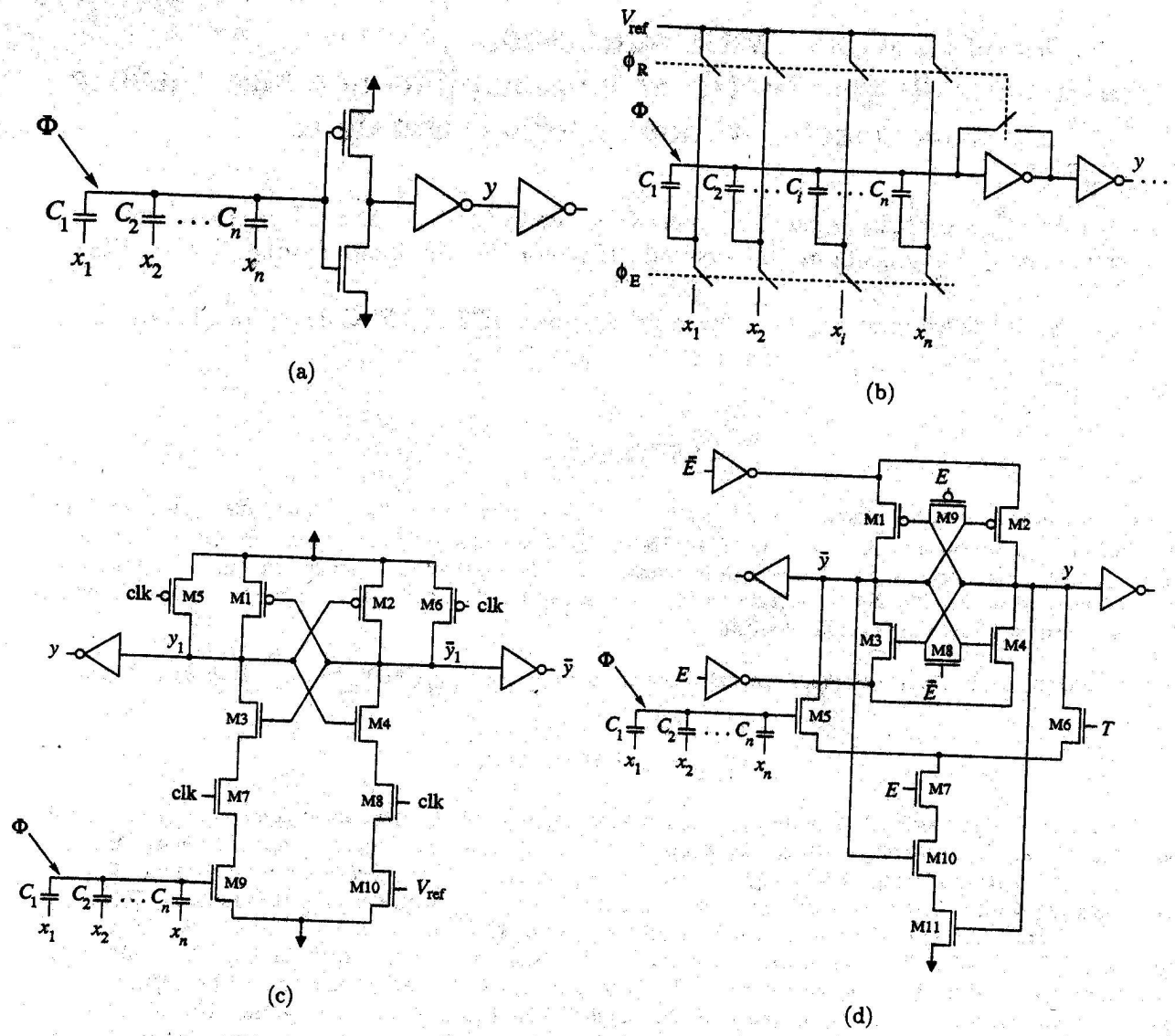

Figure 2. Voltage mode threshold gates including (a) νMOS, (b) Capacitive Threshold Logic, (c) Latched νMOS and (d) Charge Recycling Threshold Logic.

which effectively controls the threshold of the gate. The floating gate voltage, during the evaluation phase, is given by

$$\phi = V_{th} + \frac{\sum_{i=1}^{n} C_i(x_i - V_{ref})}{C_{tot}}. \tag{4}$$

Due to the reset mechanism, CTL does not require UV erasure of residual floating gate charge. The gate also has a significantly increased fan-in (up to 255[4]) compared to νMOS since the switching point variability of the primary inverter no longer influences the effective gate threshold. This is similar to the offset cancellation mechanism in chopper type comparators. The CTL gate is similar to the clocked-νMOS[2] gate, proposed at approximately the same time.

The drawbacks of the gate are that it requires a complex clocking scheme and it also suffers from high static power dissipation and low speed for similar reasons as νMOS. The gate also requires an analog reference voltage to set the threshold value which leads to difficulties in implementing CTL circuits with a large number of gates with different threshold values. The analog-reference voltage problem was overcome by the introduction of the Capacitor Programmable CTL[15] gate which requires only binary logic levels and $V_{dd}/2$ for programming.

3.1.3. Latched neuron-MOS

The Latched νMOS gate[13,16] (also referred to as Sense-Amplifier νMOS) was introduced to overcome the high power dissipation of the νMOS and CTL gates. The gate uses a current-controlled latch-sense amplifier circuit (cross coupled transistors M1-M4) instead of an inverter to perform thresholding. The gate uses the previously described νMOS structure to compute the weighted sum of inputs as illustrated in Fig. 2(c). Device parameter fluctuations are compensated by the differential amplifier configuration and the gate was shown to have significantly reduced power dissipation when compared to CTL (or clocked-νMOS) and static CMOS.[13] The fan-in is also expected to be higher than νMOS and negative weights may be implemented, due to the differential circuit structure. The threshold value may also be programmed conveniently by adding additional control capacitors. UV erasure or other technological measures must be used to remove residual floating gate charge.[16]

3.1.4. Charge Recycling Threshold Logic

Another CMOS threshold gate realization[6] has been proposed which was intended to solve the problem of high power dissipation of CTL and νMOS. The CRTL gate has low power dissipation while providing very high operating speed.[6] Fig. 2(d) shows the circuit of the Charge Recycling Threshold Logic (CRTL) gate. The sense amplifier (cross coupled transistors M1-M4) generates output y and its complement $\bar{y}$. Precharge and evaluate are 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 $\phi = \sum_{i=1}^{n} C_i x_i / C_{tot}$, 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.

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 to $V_{dd}/2$ provided that the capacitive loads at nodes y and $\bar{y}$ are equal. When E is high, the outputs are disconnected and the differential circuit (M5-M7, M10, M11) draws unequal currents from the formerly equalized nodes y and $\bar{y}$. The sense amplifier is activated after the delay of the enable inverters and amplifies the difference in potential now present between y and $\bar{y}$, 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. The advantage of equalizing the nodes y and $\bar{y}$ to $V_{dd}/2$, as opposed to precharging to Vdd as in clocked-νMOS, is that evaluation speed is increased since the cross coupled inverters begin evaluation in their transition state. Transistors M10 and M11 turn off the differential circuit after evaluation is completed to reduce the power dissipation. Negative weights may be easily implemented by using a second capacitive array on the gate of M6 to generate T. The gate was shown to reliably operate at high speed (with fan >20).[6] The gate does not have static power dissipation. The drawback of the gate is the requirement for UV erasure of residual floating gate charge. A related self-timed gate implementation has also been proposed.[17]

3.2. Conductance/Current Mode Implementations

The second class of TL gate implementations described here is based on the principle of comparing current (or conductance). Early conductance mode gates based on pseudo-nMOS or output-wired inverters[18] were fast but suffered from high power dissipation and limited fan-in. Recently, a number of high-speed differential solutions have been proposed, including Latched Comparator Threshold Logic, Equalized Current-Mode Threshold Logic and Differential Current-Switch Threshold Logic.

The underlying principle of these schemes is the use of an array of MOS transistors to implement the weighted sum of inputs in the form of a net conductance which is compared to the net conductance of a similar array of

MOS transistors used to implement the gate threshold. To obtain high speed, the comparator is implemented in the form of cross-coupled inverters. Figure 3 depicts the three current mode gates considered here.

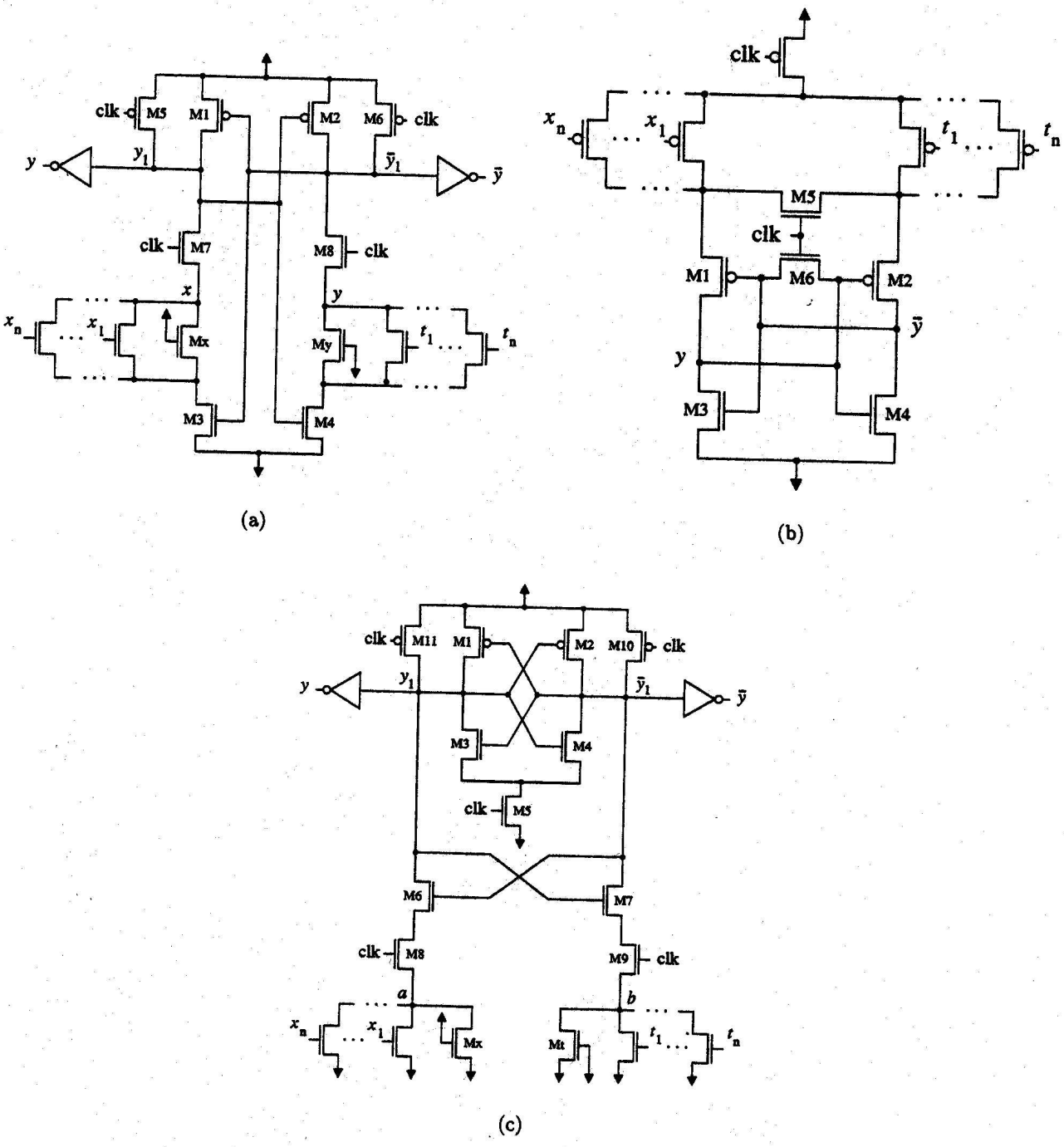

Figure 3. Current mode threshold gates including (a) Latched Comparator Threshold Logic (b) Equalized Current Mode Threshold Logic and (c) Differential Current-Switch Threshold Logic.

3.2.1. Latched Comparator Threshold Logic

The Latched Comparator Threshold Logic[3] gate shown in Fig. 3(a) consists of a current-controlled latch formed by transistors M1-M4. The input transistor array is a set of identical NMOS parallel transistors with gates connected to inputs $x_1,\ldots,x_n$. Similarly, parallel transistors with gates connected to $t_1,\ldots,t_n$ implement the threshold. Input weights are implemented by connecting each input signals to one ore more input transistor gates. Two additional transistors, Mx and My, ensure correct operation in the event that the weighted sum is equal to the threshold value.

The clock signal (clk) controls the precharge and activation of the sense circuit. The gate has two phases of operation, the precharge phase and the evaluate phase. When clk is low the nodes y_1 and $\bar{y_1}$ are precharged to V_{dd}. When clk is high, the input array and threshold setting array draw unequal currents from the precharged nodes y_1 and $\bar{y_1}$. The current-controlled latch amplifies the difference in potential now present between y_1 and $\bar{y_1}$, accelerating the transition until either y_1 or $\bar{y_1}$ reaches V_{dd}. In this way the circuit structure determines whether the net conductance of the input array is greater than or less than the net conductance of the threshold array and a TL gate is realized.

The gate does not have static power dissipation. Reliable operation with fan-in up to 20 has been reported, the gate can implement negative weights and the threshold value may be dynamically programmed. The drawback of the gate is the large capacitance at nodes x and y which slows down the sensing. To overcome this, a modified implementation in the form of Cross-coupled Inverters with Asymmetric Loads Threshold Logic[19] was proposed.

3.2.2. Equalized Current Mode Threshold Logic

The Equalized Current Mode Threshold Logic[5] gate is shown in Fig. 3(b). The gate also consists of a current-controlled latch formed by transistors M1-M4 and two banks of identical PMOS input transistors and threshold setting transistors. The important distinction is that the voltage swing is limited on the input nodes $x_1,\ldots,x_n$ and internal nodes to provide very low power dissipation on interconnects while maintaining high switching speed. The interconnect voltage is generated by a low voltage-swing interconnect driver which converts the rail-to-rail outputs y and $\bar{y}$ to a low voltage which is used to drive subsequent ECMTL gate inputs.

The clock signal (clk) controls the equalization and activation of the sense circuit. The gate has two phases of operation, the equalize phase and the evaluate phase. When clk is low the nodes y and $\bar{y}$ are equalized. When clk is high, the input array and threshold setting array draw unequal currents from V_{dd} and charge nodes y and $\bar{y}$ at different rates. The current-controlled latch regenerates the difference in potential between y and $\bar{y}$, accelerating the transition full swing outputs.

Reliable operation of ECMTL with up to 8 inputs has been reported, the gate can implement negative weights and the threshold value may be dynamically programmed. The main drawback of the gate is the requirement of a low-voltage swing driver with a separate low voltage supply to drive the gate interconnects. The gate has no static power dissipation.

3.2.3. Differential Current-Switch Threshold Logic

Finally, the Differential Current-Switch Threshold Logic[7] gate is shown in Fig. 3(c). The gate is based on the Differential Cascode Voltage Switch logic. It consists of a cross-coupled inverter latch formed by transistors M1-M4 and two banks of identical NMOS input transistors and threshold setting transistors. The clock signal (clk) controls the precharge and activation of the latch. The gate has two phases of operation, the precharge phase and the evaluate phase. When clk is low the nodes y_1 and $\bar{y_1}$ are precharged to V_{dd}. When clk is high, the input array and threshold setting array draw unequal currents and discharge nodes y_1 and $\bar{y_1}$ at different rates. The latch regenerates the difference in potential between y and $\bar{y}$, accelerating the transition to full swing outputs. Transistors M6 and M7 reduce the voltage swing on nodes a and b and cut off the static current during evaluation, reducing the power dissipation.

Reliable operation has been reported for DCSTL gates with up to 64 inputs,[20] the gate can implement negative weights and the threshold value may be dynamically programmed. The gate has no static power dissipation.

3.3. Threshold Gate Comparison

To compare the performance of the gates, the results published in the literature must be normalized. The main difficulties in providing a fair comparison are that results for the different gates are reported for various process technologies and in different circuit applications. More often than not, results presented are based on simulations alone and not measurements from fabricated circuits. This may lead to optimistic delay, power dissipation or estimated area results when layout techniques to minimize noise and reduce the influence of device parameter variation to ensure robust operation are not considered. In some instances, the results presented may also be skewed to favour a particular figure of merit (delay, area, power, power-delay product, maximum sum-of-weights etc.) without providing a complete evaluation. For these reasons the evaluation provided here is in the form of a collection of reported results in the literature, for delay and power dissipation, and a qualitative assessment of the relative advantages of each gate. From these results, conclusions may be drawn about the suitability of a particular gate for a given circuit design problem.

The focus of this section is on representative gates and small-scale circuits. Large scale arithmetic circuit applications are the subject of the next section. Table 1 summarizes the delay results reported in the literature for single gate circuits for each TL gate discussed in this work. The table gives the circuit function implemented using the given gate, the process technology used and the gate delay. In the case of the L-νMOS gate, delay is not explicitly reported, so a delay number is inferred from the reported data rate. The CTL delay numbers exclude the reset-phase time, which is on the order of 1000's of evaluation cycles.[21]

Table 1. Summary of Reported TL Gate Delay Results

Gate	Function	Process	Delay
νMOS	5-input majority[22]	0.8 μm, 5 V	0.7 ns
CTL	20-input OR[21]	0.5 μm, 5 V	5 ns
	30-input majority[4]	1.2 μm	6 ns
L-νMOS	7-input number detector[13]	0.6 μm	1ns
CRTL	4-bit carry-force[23,8]	0.35 μm, 3.3 V	250ps
LCTL	20-input majority[3]	0.7 μm, 3.3 V	3.5 ns
	31-input AND[7]	1.6 μm, 3.3 V	1.7 ns
ECMTL	8-input majority,[5] AND and OR	0.18 μm, 1.5 V	300 ps
DCSTL	31-input AND[7]	1.6 μm, 3.3 V	0.7 ns
	4-bit carry-force[23,24]	0.25 μm, 2.5 V	290 ps

Table 2 summarizes the power dissipation results reported in the literature for single gate circuits for each TL gate discussed in this work. The table shows the circuit function implemented using the given gate, the process technology used and the power dissipation result under the given conditions.

Table 2. Summary of Reported TL Gate Power Dissipation Results

Gate	Function	Process	Power Dissipation
νMOS	5-input majority[22]	0.8 μm, 5 V	280 μW (static)
CTL	20-input majority[6]	0.25 μm, 2 V	410 μW at 200 MHz data rate
L-νMOS	7-input number detector[13]	0.6 μm	400 μW at 100 MHz data rate
CRTL	20-input majority[6]	0.25 μm, 2 V	200 μW at 200 MHz data rate
LCTL	20-input majority[6]	0.25 μm, 2 V	350 μW at 200 MHz data rate
	20-input majority[3]	0.7 μm, 3.3 V	410 μW at 100 MHz data rate
ECMTL	8-input majority[5]	0.18 μm, 1.5 V	45 μW at 200 MHz data rate
DCSTL	31-input AND[7]	1.6 μm, 3.3 V	390 μW (no data rate given)

3.4. Design Considerations

A number of TL specific circuit design issues related to the mixed signal nature of the circuits must be considered. The first issue relates to the reliable operation of networks based on the gates reviewed in this work. The input offset voltage of the comparator is much more significant than the capacitor mismatch[16] in gates which use a capacitive array is to compute the weighted sum of inputs. Conversely, the current mode gates are susceptible to the significant input and threshold setting transistor mismatch. In addition, to maintain a symmetrical load on in the comparator circuit, dummy transistors[7] may have to be used which do not perform a computational function and increase the gate area. Each of the gates must be considered as an analog circuit requiring the use of known layout techniques to match devices and minimize the impact of noise. These include substrate voltage control, shielding, isolation, same-orientation layout of transistors and the use of small parallel transistors to realize larger devices with reduced statistical parameter variations. It is also worth mentioning that various techniques can can be used for designing reliable circuits based on unreliable gates, including coding,[31] hardware redundancy[32] and synthesis techniques which introduce don't care conditions.

4. THRESHOLD LOGIC APPLICATIONS

To provide a perspective on the attainable performance of digital systems designed using threshold logic, Table 3 provides a collection of results on the design of digital circuits reported in the literature. The table includes reported measured and simulated results and shows the comparison to the equivalent circuits designed using conventional CMOS logic. Where available, the reported comparison is of the given circuit relative to conventional CMOS logic (either dynamic-CMOS or static-CMOS). For example, the νMOS full adder was shown to occupy 45% less area and was 30% slower than the static-CMOS implementation.[25] To the best of our knowledge, no ECMTL applications have been reported. Where the reported results are based on simulations or fabricated chip measurements, this is denoted in the table by "(meas.)", all other results are based on circuit or extracted layout simulations. The table shows that TL potentially offers great advantages compared to conventional logic in terms of area, power or delay across a wide range of digital circuit designs.

Table 3. Summary of Reported TL Gate Applications

Gate	Function	Process	Results	CMOS Comparison
νMOS	5-input majority[22]	0.8 μm	390 μm^2	60% less area, 17% slower
	Full adder[25]	2 μm	831 μm^2 area (meas.)	45% less area, 30% slower
	(15,4) counter using νMOS sorter[26]	1.2 μm	8 ns delay	28% faster
	8-input Muller-C element[27]	0.8 μm	4930 μm^2, 1.8 ns delay, 30 μW at 100 MHz (meas.)	50% less area, 44% faster, 94% lower power
	(4:2) compressor[28]	0.8 μm	1.2 ns delay	61% faster, 20% lower power delay product
	(6:2) compressor[28]	0.8 μm	1.6 ns delay	64% faster, 22% lower power delay product
CTL	(31,5) counter[9]	1.2 μm	4.2 ns delay, 80000 μm^2 (meas.)	approx. 50% less area
	(8×8)-bit multiplier[9]	1.2 μm	70000 μm^2, up to 30 MHz data rate (meas.)	(no comparison provided)
CRTL	(7,3) shared-capacitor counter[29]	0.35 μm	460 ps delay, 70 trans., 1850 μm^2	6% area reduction, 45% faster
	(15,4) shared-capacitor counter[29]	0.35 μm	480 ps delay, 140 trans., 3960 μm^2	(no comparison provided)
	4-bit adder[11]	0.25 μm	2280 μm^2, 104 trans., 500 μW at 100 MHz	41% less area, 75% lower power
	64-bit hybrid CRTL/CMOS adder[8]	0.35 μm	670 ps critical path delay, 4325 trans.	30% faster, 30% fewer trans.
LCTL	16-bit adder[30]	1 μm	126000 μm^2, 11 ns delay, 37 μW at 3.3V, 100 MHz, 1328 trans.	(no comparison provided)
DCSTL	(7,3) hybrid DCSTL/CMOS counter[10]	0.25μm	345 ps delay, 237 trans.	53% faster, 67% more trans.
	Mod. DCSTL 16-input embedded TL flip-flop[20]	0.25 μm	1470 μm^2, 1.0 to 2.1 ns (worst case) delay (meas.)	(no comparison provided)

5. CONCLUSIONS

A review of the significant recent developments in TL has been presented, including gate implementations and applications in computer arithmetic. The summary of applications has shown that the reviewed TL gates are suitable for the design of high performance digital circuits with reduced area, power dissipation and delay as shown in Table 3. The advantages of TL are essentially a matter of increased efficiency. Area or power reduction leads to reduced cost, or alternatively, increased functionality for the same power or cost.

Threshold logic, however, continues to remain almost exclusively the subject of research work. A large number of publications claim that a particular TL based circuit or gate reduces power/delay/area by a certain percentage relative to conventional CMOS, and it is our belief that more work of this nature will not contribute to the incorporation of TL techniques in industrial applications. The authors believe that in order for TL to gain industry acceptance, a number of large scale system designs based on TL must be able to demonstrate significant advantages compared to state-of-the-art conventional CMOS logic designs. There is evidence that a hybrid conventional-logic/TL approach[8,10] could help achieve this. Additionally, lack of high-level synthesis tools for TL design must also be resolved.

ACKNOWLEDGMENTS

The support of the Australian Research Council, the Delft University of Technology and the Sir Ross and Sir Keith Smith Fund is gratefully acknowledged.

REFERENCES

1. T. Shibata and T. Ohmi, "An intelligent MOS transistor featuring gate-level weighted sum and threshold operations," in *IEDM, Technical Digest*, IEEE, (New York, NY, USA), December 1991.
2. K. Kotani, T. Shibata, M. Imai, and T. Ohmi, "Clocked-neuron-MOS logic circuits employing auto-threshold-adjustment," in *ISSCC Digest of Technical Papers*, pp. 320–321, 1995.
3. M. Avedillo, J. Quintana, A. Rueda, and E. Jiménez, "Low-power CMOS threshold-logic gate," *IEE Electronics Letters* **31**, pp. 2157–2159, December 1995.
4. H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, and U. Çiliniroğlu, "A capacitive threshold-logic gate," *IEEE JSSC* **31**, pp. 1141–1149, August 1996.
5. S. Bobba and I. Hajj, "Current-mode threshold logic gates," in *Proceedings of the IEEE International Conference on Computer Design*, pp. 235–240, (Austin, Texas, USA), 2000.
6. 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**, pp. 1067–1069, August 2001.
7. M. Padure, S. Cotofana, and S. Vassiliadis, "A low-power threshold logic family," in *Proc. IEEE International Conference on Electronics, Circuits and Systems*, pp. 657–660, 2002.
8. 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)*, To be published.
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**, pp. 1177–1183, August 1996.
10. 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*, pp. 457–460, 2002.
11. 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*, pp. 469–472, (Phoenix), May 2002.
12. T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE JSSC* **39**, pp. 1444–1455, 1992.

13. K. Kotani, T. Shibata, M. Imai, and T. Ohmi, "Clock-controlled neuron-mos logic gates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **45**, pp. 518–522, April 1998.
14. U. Cilingiroglu, "A purely capacitive synaptic matrix for fixed-weight neural networks," *IEEE Transactions on Circuits and Systems* **38**, pp. 210–217, February 1991.
15. A. Stokman, S. Cotofana, and S. Vasilliadis, "A versatile threshold logic gate," in *CAS '98 Proceedings*, pp. 163–166, October 1998.
16. A. Luck, S. Jung, R. Brederlow, R. Thewes, K. Goser, and W. Weber, "On the design robustness of threshold logic gates using multiple-input floating gate MOS transistors," *IEEE Transactions on Electron Devices* **47**, pp. 1231–1239, June 2000.
17. P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott, "Compact parallel (m,n) counters based on self timed threshold logic," *IEE Electronics Letters* **38**, pp. 633–635, June 2002.
18. K. J. Schultz, R. J. Francis, and K. C. Smith, "Ganged CMOS: Trading standby power for speed," *IEEE JSSC* **25**, pp. 870–873, 1990.
19. J. F. Ramos, J. A. H. Lopez, M. J. Martin, J. C. Tejero, and A. "A threshold logic gate based on clocked coupled inverters," *International Journal of Electronics* **84**(-), pp. 371 -82, 1998.
20. M. Padure, S. D. Cotofana, and S. Vassiliadis, "Design and ... ental results of a CMOS flip-flop featuring embedded threshold logic," in *Proc. IEEE International Symposium on Circuits and Systems*, May. To be published.
21. M. Padure, S. D. Cotofana, M. Bodea, and S. Vassiliadis, "Capacitive threshold logic: a designer perspective," in *CAS '99 Proceedings*, pp. 81 –84, 1999.
22. R. Lashevsky, K. Takaara, and M. Souma, "The efficiency of neuron-MOS transistors in threshold logic," *Soft Computing* **3**(1), pp. 20–29, 1999.
23. S. Vassiliadis, S. Cotofana, and K. Bertels, "2-1 addition and related arithmetic operations with threshold logic," *IEEE Trans. Computers* **45**, pp. 1062–1067, September 1996.
24. M. Padure, S. D. Cotofana, S. Vassiliadis, C. Dan, and M. Bodea, "Compact delay model for latch-based threshold logic gates," in *Proceedings of the IEEE International Semiconductor Conference*, vol. 2, pp. 317–320, 2002.
25. K. Hirose and H. Yasuura, "A comparison of parallel multipliers with neuron MOS and CMOS technologies," in *Proceedings of IEEE Asia Pacific Conference on Circuits and Systems 96*, pp. 488–491, IEEE, November 1996.
26. E. Rodriguez-Villegas, M. Avedillo, J. Quintana, G. Huertas, and A. Rueda, "A νMOS based sorter for arithmetic applications," *VLSI Design* **11**(2), pp. 129–136, 2000.
27. E. Rodriguez-Villegas, G. Huertas, A. M. J., Q. J. M., and R. A., "A practical floating-gate muller-c element using vMOS threshold gates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **48**, pp. 102–106, January 2001.
28. J. M. Quintana, M. J. Avedillo, E. Rodriguez-Villegas, and A. Rueda, "Threshold-logic-based design of compressors," in *Proc. International Conference on Electronics, Circuits and Systems*, vol. 2, pp. 661–664, 2002.
29. P. Celinski, S. D. Cotofana, and D. Abbott, "Area efficient, high speed parallel counter ... sing charge recycling threshold logic," in *Proc. IEEE International Symposium on Circuits an... ..., 2005.* published.
30. J. F. Ramos, J. C. Tejero, J. A. Hidalgo, M. J. Martin, a... ..., two operand bin... adde... ... threshold logic," in *Proceedings DCICS'98*, (Madrid), 1998.
31. R. Hedge and N. R. Shanbhag, "Energy-efficiency in presence of deep ...bmicron noise," in *Proceedings of ICCAD*, November 1998.
32. S. Muroga, *Threshold Logic and Its Applications*, Wiley, New York, 1971.