AREA EFFICIENT, HIGH SPEED PARALLEL COUNTER CIRCUITS USING CHARGE RECYCLING THRESHOLD LOGIC

Peter Celinski, Derek Abbott

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

Sorin D. Cotofana

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 of a low depth, highly compact implementation of parallel counters (i.e., population counters), based on threshold logic. Two such counters are designed using the recently proposed Charge Recycling Threshold Logic (CRTL) gate. The novel feature of the designs is the sharing among all threshold gates of a single capacitive network for computing the weighted sum of all input bits. The significance of the result is the reduction by almost 35% in the required number of capacitors for the (7,3) counter and by over 60% for the (15,4) counter. This reduces the total area by approximately 37% for the (7,3) counter and by 60% for the (15,4) counter, with no increase in delay. The proposed (7,3) counter design is also shown to be 45% faster compared to a conventional Boolean full-adder based circuit.

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, area efficiency and functionality of the arithmetic and other units contained therein. Low power dissipation has become a major issue demanded by the high performance processor market in order to meet the high density requirements of advanced VLSI processors.

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. 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, and a number of applications based on TL gates have demonstrated its ability to achieve high operating speed and significantly reduced area [1, 2, 3, 4].

The aim of this paper is to develop highly compact parallel counter circuits using TL. We begin in Section 2 by giving a brief overview of threshold logic. This is followed by a description of Charge Recycling Threshold Logic (CRTL) in Section 3. Section 4 gives an overview of threshold logic counters, followed by a description of the the proposed counter circuits in Section 5. Finally, a brief conclusion is given in Section 6.

2. THRESHOLD LOGIC

A threshold logic gate is functionally similar to a hard limiting neuron. 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. A linear weighted sum of the binary inputs is computed followed by a thresholding operation.

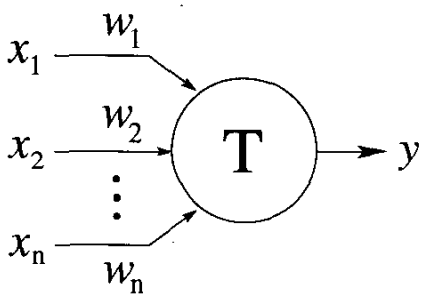

Figure 1: Threshold Gate Model

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 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 $T = n$ 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 suggested by a number of practical results [5, 6, 7] which suggest advantages of TL over conventional Boolean logic.

0-7803-7761-3/03/$17.00 ©2003 IEEE

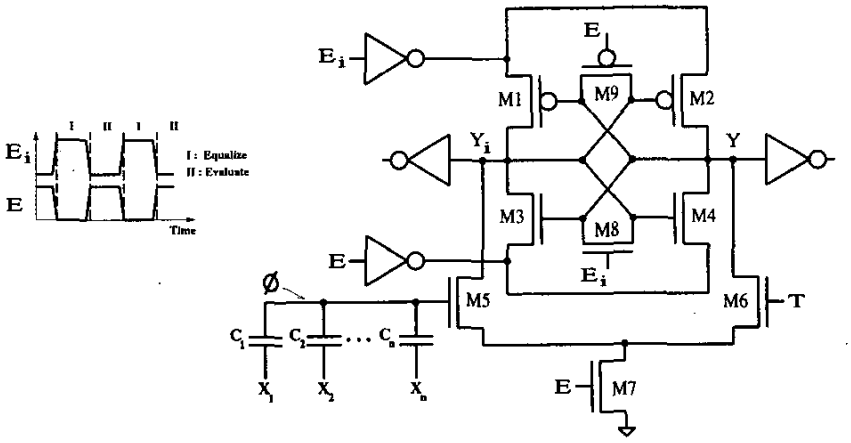

Figure 2: The proposed CRTL gate structure

3. CHARGE RECYCLING THRESHOLD LOGIC

We now briefly describe the realization for CMOS threshold gates presented in [3]. This TL gate implementation is used in the proposed counter circuits. Fig. 2 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 realised by setting capacitors C_i to appropriate values. Typically, these capacitors are 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. 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. It must be noted that the voltage T can also be replaced by a set of capacitively coupled inputs onto the floating gate of M6, thus realizing negatively weighted inputs. The gate was shown to reliably operate at high speed (> 600 MHz in a 0.35 μm process) and had typically around 15 to 20% lower power dissipation than other capacitive CMOS TL gate implementations [3].

4. THRESHOLD LOGIC PARALLEL COUNTERS

An (m,n) binary counter is a combinatorial network which generates a binary coded output vector of length n which corresponds to the number, or count, of logic 1's in the m-bit input vector. Usually $n = \log_2(m+1)$ and such counters are referred to as saturated. The full adder is a particular case of a counter with 3 inputs and 2 outputs, thus it is a (3,2) counter. Counters are important in various applications, the most common of which are the reduction of the partial product tree in parallel multipliers and the realization of multiple-input adders.

The concept of parallel counters was originally proposed by Dadda [8] who also developed a scheme for interconnecting small counters to design counters with a larger number of inputs. In conventional logic, higher order counters, such as (7,3), (15,4) or (31,5), have traditionally been implemented by using trees of (3,2) counters because of the disadvantages of a direct implementation. However, counters consisting of such full adder trees have a relatively high delay and grow rapidly with input vector size in terms of the required number of full adders. Threshold logic allows us to design area efficient counters which operate at higher speeds than the same counters built using trees of full adders.

The natural way to implement counters is to utilize known networks devised for computing symmetric functions such as parity. These include designs by Muroga [9], Kautz [10] and Minnick [11]. The Minnick scheme based counter worst case delay is equal to two TL gates, and it provides a good area-delay tradeoff compared to the other designs. For these reasons we have chosen the Minnick scheme as the basis for our proposed counter circuits. The (7,3) and (15,4) counters will be used as illustrative examples, but the technique presented here can be equally applied to higher order counters.

Fig. 3 shows the truth table and TL network for the (7,3) counter. The input v consists of the seven input bit lines, each having a weight of 1, and is denoted by a thick black line to differentiate it from the single bit lines. Fig. 4 shows the TL network for the (15,4) counter based on Minnick's scheme.

5. PROPOSED PARALLEL COUNTER CIRCUITS

The (7,3) counter consists of 5 threshold gates arranged in two layers, three in the first layer and two in the second layer. The circuit diagram showing the proposed design is shown in Fig. 5. Usually the implementation of this counter using one of the recently proposed capacitive TL gates would require a network of 7 capacitors connected to the 7 inputs to be duplicated at each of the gates in the circuit. Therefore, significant area would be associated with

v	y_2	y_1	y_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

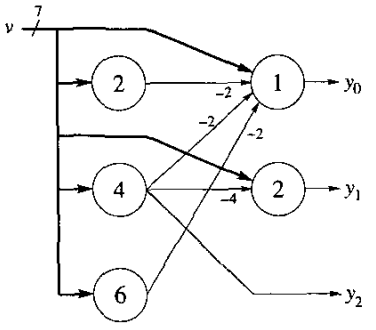

Figure 3: The (7,3) counter truth table and the Minnick TL network

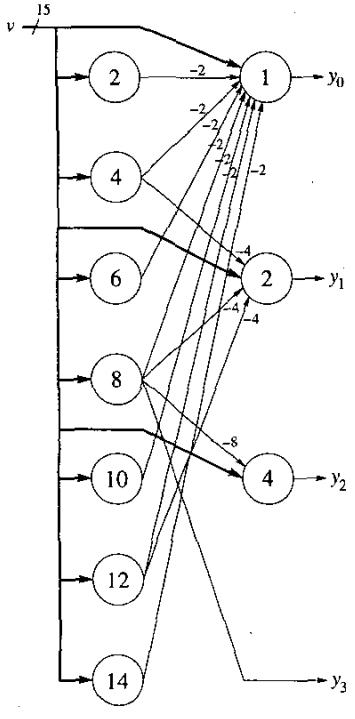

Figure 4: The (15,4) Minnick scheme based counter TL network

routing the 7 interconnect lines to each of the 5 gates, and with duplicating the 7 capacitors at each gate. These drawbacks have an even greater impact on total area for higher order counters. The innovation proposed here is to implement the capacitive network which calculates the analog value of the sum of the counter input bits only once, and this value becomes one input of the sense amplifier in each CRTL gate. In the second layer gates, the other sense amplifier input is connected to the capacitive networks which implement the negative weights of the layer 1 to layer 2 interconnections. Such an arrangement is made possible by the differential nature of the CRTL gate, and is possible using other differential capacitive TL gates. Output y_2 is available after one gate delay and the remaining outputs after two gate delays. All outputs can be made to evaluate simultaneously by adding one additional CRTL gate which would act as a delay element for y_2. Fig. 6 shows the circuit corresponding to the (15,4) counter shown in Fig. 4.

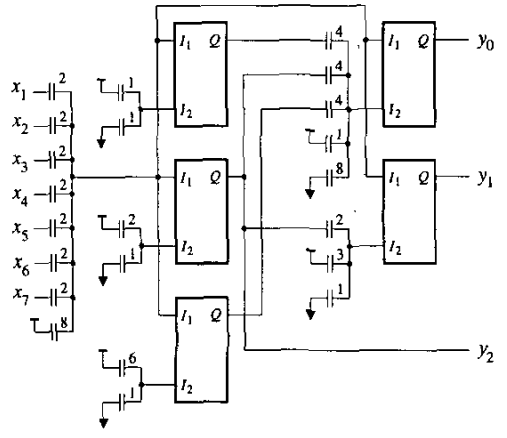

Figure 5: Circuit diagram of the proposed CRTL Minnick scheme based (7,3) counter

The numbers next to the capacitors indicate the multiple of the unit capacitor. Enable signal inputs are not shown to improve clarity. The two gates in the second layer are enabled after the outputs from the first layer are evaluated. Capacitors shown connected to Gnd and V_{dd} adjust the effective threshold of each CRTL gate. The inputs denoted by I_1 and I_2 in Figs. 5 and 6 correspond to the ϕ and T inputs, respectively, shown in Fig. 2.

Compared to the design of the (7,3) counter based on Minnick's scheme which does not use capacitor sharing, the required number of (unit) capacitors has been reduced from 94 to 61, and for the (15,4) counter the required number of (unit) capacitors has been reduced from 384 to 152.

To evaluate the area savings, the (7,3) and (15,4) counters were laid out using the proposed capacitor sharing technique (SCRTL) and without sharing (CRTL) using a 3.3V, double-poly, 4-metal 0.35 μm process. For comparison, a (7,3) counter using the traditional Boolean full-adder (BL) based scheme [12] was also laid out in the same process. The extracted layouts were simulated using HSPICE. Table 1 shows the area and delay results. For the CRTL based counters, delay refers to the latency, since the gates are clocked.

As is shown in the table, the CRTL based designs are approximately 45% faster than the full-adder implementation for the (7,3) counter. The CRTL (7,3) counter without sharing occupies 50% more area than the full-adder design, and the capacitor-sharing technique provides a small area saving when compared to the full-adder based counter. The sharing of capacitors reduces the area by 37% for the (7,3) counter and by 60% for the (15,4) counter compared to the non-shared designs. It should be noted that the use of capacitor sharing does not result in an increase in delay.

A number of other circuit designs have been proposed in the literature based on different process technologies. These include the (15,4) counter using a neuron-MOS based sorter circuit [13] which was reported to have a delay of 8ns in a 1.2μm process and for which no area data was given. The design for a (31,5) counter using Capacitive Threshold Logic (CTL) was presented in [5], the worst case delay was 4.2ns in a 1.2μm process and the area was

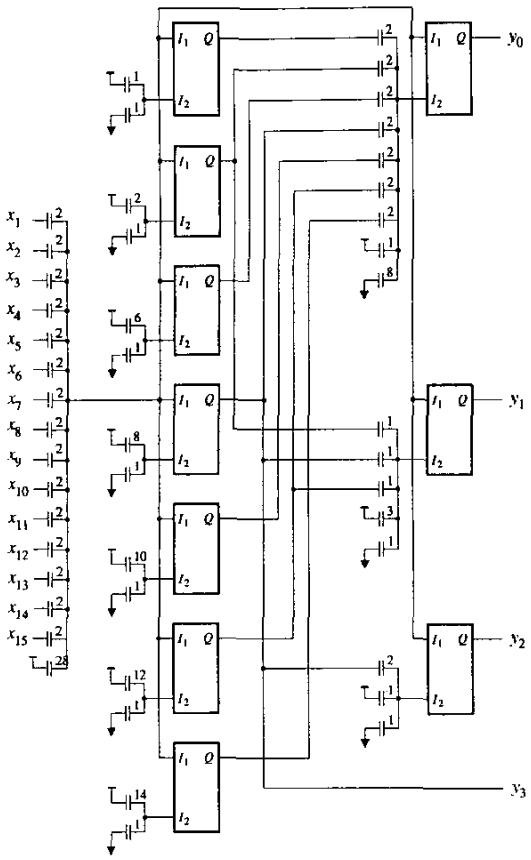

Figure 6: Circuit diagram of the proposed CRTL Minnick scheme based (15,4) counter

approximately 80000μm^2. Recently a hybrid Threshold-Boolean logic (HTBL) design was proposed [6] which uses a single-phase clocked differential cascode voltage switch circuit implementation of the TL gate in combination with static CMOS logic. The delay of the (7,3) HTBL counter in a 0.25μm process was 345ps and the design required 237 transistors. An accurate comparison would require these to be redesigned using the 0.35μm process used in this work, which is beyond the scope of this paper.

6. CONCLUSIONS

A highly compact implementation of parallel counters based on the Minnick network has been proposed. Two such counters were designed based on the proposed technique using the CRTL gate. The result was the reduction by almost 35% in the required number of capacitors and for the (7,3) counter, and by over 60% in number of capacitors for the (15,4) counter. The capacitive sharing resulted in a reduction by 37% in area compared to the design which does not use sharing for the (7,3) counter. Compared to the traditional full-adder based design for the (7,3) counter, the proposed CRTL designs were shown to be 45% faster.

Table 1: Counter comparison (0.35μm 2P/4M process)

Counter Type	Delay (ps)	# Tran.	# Unit Caps.	Area (μm^2)	Norm. Area
BL (7,3) [12]	840	136	-	1970	1
CRTL (7,3)	460	70	94	2952	1.5
SCRTL (7,3)	460	70	61	1848	0.94
CRTL (15,4)	480	140	384	10124	1
SCRTL (15,4)	480	140	152	3956	0.4

7. REFERENCES

[1] K. Kotani, T. Shibata, M. Imai, and T. Ohmi, "Clocked-neuron-MOS logic circuits employing auto-threshold-adjustment," in *ISSCC Digest of Technical Papers*, 1995, pp. 320–321.

[2] H. Özdemir, A. Kepkep, B. Pamir, Y. Leblebici, and U. Çiliniroğlu, "A capacitive threshold-logic gate," *IEEE JSSC*, vol. 31, no. 8, pp. 1141–1149, August 1996.

[3] P. Celinski, J. F. López, S. Al-Sarawi, and D. Abbott, "Low power, high speed, charge recycling CMOS threshold logic gate," *IEE Electronics Letters*, vol. 37, no. 17, pp. 1067–1069, August 2001.

[4] M. Padure, S.D. Cotofana, and S. Vassiliadis, "A low-power threshold logic family," in *Proc. IEEE International Conference on Electronics, Circuits and Systems*, 2002, pp. 657–660.

[5] 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*, vol. 31, no. 8, pp. 1177–1183, August 1996.

[6] M. Padure, S. Coofana, 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*, 2002, pp. 457–460.

[7] 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*, Phoenix, May 2002, pp. 469–472.

[8] L. Dadda, "Some schemes for parallel multipliers," *Alta Freq.*, vol. 34, pp. 349–355, May 1965.

[9] S. Muroga, *Threshold Logic and Its Applications*, Wiley, New York, 1971.

[10] W. H. Kautz, "The realization of symmetric switching functions with linear input logical elements," *IRE Transactions on Electronic Computers*, vol. EC-10, pp. 371–378, March 1961.

[11] R. C. Minnick, "Linear-Input Logic," *IRE Transactions on Electronic Computers*, vol. EC-10, pp. 6–16, March 1961.

[12] I. Koren, *Computer Arithmetic Algorithms*, A. K. Peters, Natick, MA, 2002.

[13] E. Rodriguez-Villegas, M.J. Avedillo, J.M. Quintana, G. Huertas, and A. Rueda, "A νMOS based sorter for arithmetic applications," *VLSI Design*, vol. 11, no. 2, pp. 129–136, 2000.