A Low Power, High Speed Threshold Logic and its Application to the Design of Novel Carry Lookahead Adders

Peter Celinski[a] Jose F. López[b] S. Al-Sarawi[a] and Derek Abbott[a]

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

[b]Research Institute for Applied Microelectronics
Universidad de Las Palmas de G.C., 35017
Spain.

ABSTRACT

The first main result of this paper is the development of a low power threshold logic gate based on a capacitive input, charge recycling differential sense amplifier latch. The gate is shown to have very low power dissipation and high operating speed, as well as robustness under process, temperature and supply voltage variations. The second main result is the development of a novel, low depth, carry lookahead addition scheme. One such adder is also designed using the proposed gate.

Keywords: threshold logic, charge recycling, low power, carry lookahead addition

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. The importance of low power is also evident in portable and aerospace applications, and is related to issues of reliability, packaging, cooling and cost.

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 traditional AND-OR-NOT (AON) logic-gate based design. However, lack of efficient physical realizations has meant that TL has, until recently, 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]

Both static and dynamic TL gate implementations have been devised. Purely static gates such as neuron-MOS suffer from limited fan-in,[1] typically less than 12 inputs, and also low speed.[2] Also, some of the existing dynamic gates have relatively high static power dissipation, and some require multiple clock phases[1],[3] introducing the drawbacks associated with clock signal routing cost, clock skew and clock power dissipation. Although the non-capacitive dynamic approach[4] dissipates no static power, its dynamic power dissipation is comparable to the total power dissipation of other existing approaches.

We begin in Section 2 by giving a brief overview of threshold logic (TL). This is followed by a description of the proposed Charge Recycling Threshold Logic[5] in Section 3 and its performance evaluation in Section 4. In Section 5 the proposed carry lookahead addition scheme is presented, and the designs for two novel CLAs are shown in Section 6. Finally a brief conclusion is given in Section 7.

Electronics and Structures for MEMS II, Neil W. Bergmann, Derek Abbott, Ahsan J. Hariz, Vijay K. Varadan, Editors, Proceedings of SPIE Vol. 4591 (2001) © 2001 SPIE · 0277-786X/01/$15.00

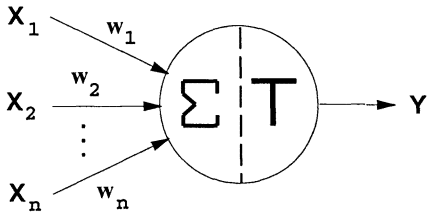

Figure 1. Threshold Gate Model

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.

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 ith input variable X_i.[6] The output Y is given by

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

This may be written in a more compact form using the sgn function as

$$Y = \text{sgn}\left(\sum_{i=1}^{n} w_i X_i - T \right). \tag{2}$$

The sgn function is defined as $\text{sgn}(x) = 1$ if $x \geq 0$ and $\text{sgn}(x) = 0$ if $x < 0$. Any threshold function may be computed with positive integral weights and a positive real threshold, and all Boolean functions can be realized by a threshold gate network of depth at most two.[6] 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. Despite the large body of work which exists on the theory of threshold networks, mainly within the framework of computational circuit complexity, relatively little work has been carried out on the implementation of TL systems in silicon.[7]

3. CHARGE RECYCLING THRESHOLD LOGIC (CRTL)

We now describe a new realization for CMOS threshold gates which operates on a single phase clock, is capable of high speed operation, is suitable for high fan-in gate implementation and has a very low overall power dissipation. The proposed Charge Recycling Threshold Logic (CRTL) gate is latched which removes the requirement for an additional latch to synchronise outputs. This makes it highly suitable for pipelined operation, as is demonstrated in the proposed design for a pipelined carry lookahead adder in Section 6.

Fig. 2 shows the proposed circuit structure for implementing a threshold gate with positive weights and threshold. It is based on the charge recycling Asynchronous Sense Differential Logic (ASDL) developed by Bai-Sun *et al.*[8] The main element is the sense amplifier (cross coupled transistors M1-M4) which generates output Y and its complement Y_i. Precharge and evaluate is specified by the dual Enable clock signals 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, although alternatives, such as trench capacitors available in DRAM processes, or MIM capacitors available in some processes, may obviously also be used.

The ASDL comparator architecture from which the proposed CRTL gate is derived implements high performance, energy efficient operation by re-using the charge which was drawn from the supplies during evaluation, in the equalization phase. The enable signal E controls the precharge and activation of the sense circuit. Transistors M8

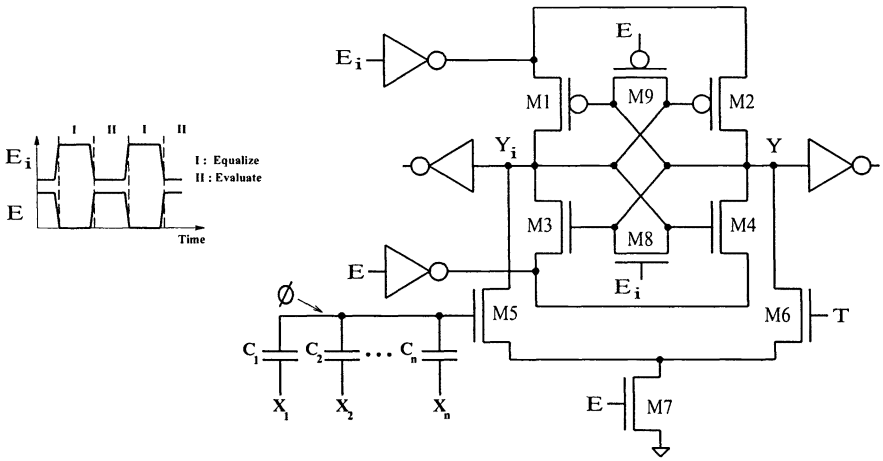

Figure 2. The proposed CRTL gate structure

and M9 equalize the outputs. The logic 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.

To ensure reliable operation, the gate layout must be symmetrical to minimize the transistor mismatches and interconnects must be of similar length and width to eliminate interconnect-related mismatch. The delay of the enable inverter must be sufficiently large so that the output nodes have sufficient voltage difference at the start of sensing to overcome any offset voltage present in the cross coupled sense amplifier.

4. CRTL PERFORMANCE EVALUATION

To evaluate and compare the performance of the proposed CRTL gate against other CMOS TL gate implementations, a 20-input majority gate (T=10, achieved by setting voltage $T=V_{dd}/2$) was designed in an industrial 0.25 μm process. The 20-input majority function was also implemented using clocked neuron-MOS,[1] CMOS Capacitor Coupling Logic (CCCL)[3] and the non-capacitive TL structure reported in[4] (LPTL). The unit capacitance value used in each of the capacitive gate implementations was 5 fF. To compare the power dissipation, each of the gates was designed to have similar delay, output rise and fall times, and was loaded by equally sized inverters. All transistors were of minimum length for each implementation and transistor widths were selected to achieve the above timing requirements. All inputs to each gate were switched such that during each evaluation cycle the minimum majority or minority was achieved (11 out of 20 inputs were high or low, respectively). Also, the power dissipated in the inverters driving the clock and data inputs was included in the total power dissipation measured for each gate. Fig. 3 shows the HSPICE power dissipation simulation results for each of the gates versus operating frequency for a 2 V supply. As shown in the Figure, at a typical operating frequency of 200 MHz, CRTL improves power dissipation by between 15% and 30% over the other CMOS threshold gate implementations.

To ensure correct behavior under process and operating point variations, the proposed gate was tested at 45 corners (V_{dd} at 2 V, 2.5 V and 3 V, process Slow-Slow, Slow-Fast, Fast-Slow, Fast-Fast and Typical-Typical, and temperature at -25^{o}C, 75^{o}C and 125^{o}C). Fig. 4 shows the transient waveform results from the HSPICE simulation for the 2V-typical-75^{o}C corner at 300 MHz data rate. Simulation results of the 20-input majority gate also indicate that the CRTL gate can operate even at frequencies over 400 MHz with low power dissipation (below 400 μW) under worst case conditions (Vdd=2 V, 125^{o}C, Slow-Slow transistor corner).

5. CARRY LOOKAHEAD ADDITION WITH THRESHOLD LOGIC

Addition is one of the most critical operations performed by VLSI processors. Adders are used in ALUs, floating-point arithmetic units, memory addressing and program counter updates. The critical requirement of the adder is speed,

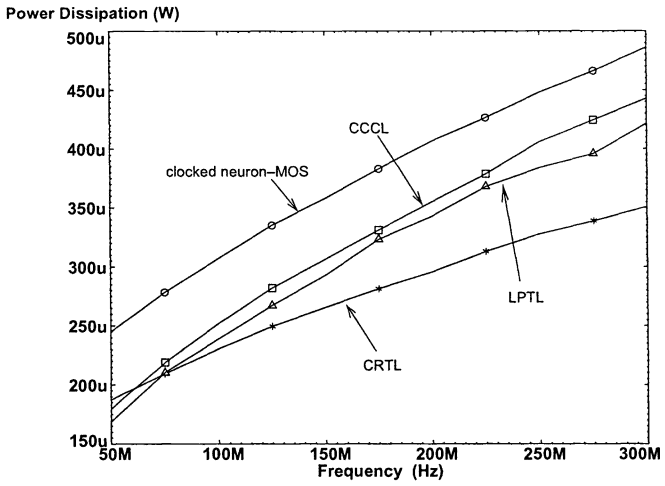

Figure 3. Power Dissipation vs. Frequency comparison

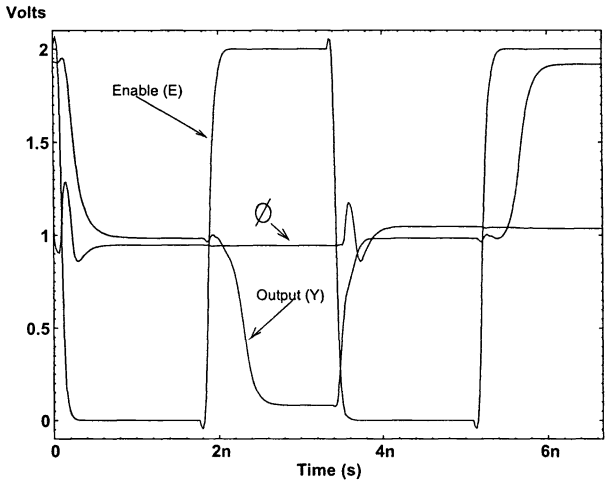

Figure 4. Input, Enable and Output simulation results

but low power dissipation and area efficiency have become increasingly important in recent years. The key factor in the proposed addition scheme is the introduction of high-valency threshold logic carry generate and propagate cells, which results in reduced logic depth addition networks, and hence reduced area and power dissipation.

Carry lookahead is a well-known technique for decreasing the latency of addition by reducing the logic depth to $O(log_2 w)$, where w is the wordlength of the addends. It is one of the fastest addition algorithms, and allows significant design trade-offs to be made in terms of latency, area and power. The addition problem can be expressed in prefix notation in terms of generate (g_j), propagate (p_j) and carry (c_j) signals at each bit position j for a width, w adder with the following equations:

$$g_j = a_j.b_j \tag{3}$$

$$p_j = a_j + b_j \tag{4}$$

$$c_j = g_j + \sum_{i=0}^{j-1} \left(g_i . \prod_{k=i+1}^{j} p_k \right) \tag{5}$$

$$s_j = c_{j-1} \oplus a_j \oplus b_j, \tag{6}$$

where $j = 0, \ldots, w-1$, c_j denotes the carry generated at position i and c_{-1} denotes the carry into the LSB position. From these expressions the block-generate and block-propagate signals G_j^i and P_j^i can be written as:

$$G_j^i = \begin{cases} a_j.b_j, & \text{for} \quad i = j \\ G_j^k + P_j^k.G_{k-1}^i, & \text{for} \quad j \geq k > i \end{cases} \tag{7}$$

$$P_j^i = \begin{cases} a_j + b_j, & \text{for} \quad i = j \\ P_j^k.P_k^i, & \text{for} \quad j \geq k > i. \end{cases} \tag{8}$$

Assuming that $c_{-1} = 0$, then the carry signal at position j, c_j, is given by:

$$c_j \;=\; G_j^0. \tag{9}$$

If we let GP_j^i represent the pair $(G_j^i,\ P_j^i)$, then the above expression can be written using the Brent-Kung $\bullet$ operator as

$$GP_j^i \;=\; (G_j^k + P_j^k G_{k-1}^i,\ P_j^k P_{k-1}^i) \;=\; GP_j^k \bullet GP_{k-1}^i. \tag{10}$$

The direct approach to implementing this scheme in, for example, static CMOS is not practical for any useful wordlength $w \geq 16$, since the amount of circuitry required to assimilate the MSB carry becomes prohibitive. For this reason, and also because of the associative nature of the expressions for g_i and p_i, carry lookahead adders are usually built using a parallel tree structure. The threshold logic approach can, however, be used to design circuits which implement carry lookahead addition in a more direct and efficient way than the static CMOS prefix-tree approach.

The $\bullet$ prefix cell defined by Equation (10) operates on two input signal pairs (GP_j^k and GP_{k-1}^i) and produces the signal pair GP_j^i. For this reason it is referred to as having a valency of 2. A simplified prefix cell replaces the $\bullet$ prefix cell at the final cell position in the prefix tree before the sum is calculated as only the group generate signal is required to evaluate the carry signal at each bit position. Beaumont-Smith *et al*[9] recently generalised this concept by introducing higher valency prefix cells for the first time. By using rows of higher valency prefix cells, it was shown that it is possible to significantly reduce the number of cells in the critical path or shorten interconnect lengths, since the number of carries assimilated at each level in the tree is increased, at the expense of increased prefix cell delay. Typically, in a sub-micron CMOS process the fan-in and hence maximum valency is limited to 4.

A modified set of the Boolean Equations (3)-(10) will now be derived in a form suitable for implementation in threshold logic. We will take advantage of the high fan-in capability of TL to design high valency prefix-cells, that is, prefix-cells which compute group propagate, group generate and carry signals from a large number of input bits.

Instead of computing the bit-wise generate and propagate signals ($g_i,\ p_i$), the input operands ($a_i,\ b_i$), $i = 0, \ldots, w-1$, are grouped into n-bit blocks. The first stage starts with the computation of the group-generate and group-propagate signals (G_j^{j-n+1} and P_j^{j-n+1}) for each block, directly from the input operands. A carry is generated in a group if the sum of the n bits in the group exceeds (is strictly greater than) the maximum number representable by n sum bits. Therefore a group-propagate signal G_j^{j-n+1} is 1 if the sum of the n bits in the group exceeds the maximum number representable by n sum bits. Similary, the group propagates a carry originating in the neighbouring group of lower significance and the group propagate signal P_j^{j-n+1} is 1 if the sum of the n bits in the group is equal to or greater than the maximum number representable by n sum bits. This may be written in general equation form as:

$$G_j^{j-n+1} = \text{sgn}\left(\sum_{k=j-n+1}^{j} 2^{k-(j-n+1)}(a_k + b_k) - (2^{n+1} - 1) \right) \tag{11}$$

$$P_j^{j-n+1} = \text{sgn}\left(\sum_{k=j-n+1}^{j} 2^{k-(j-n+1)}(a_k + b_k) - (2^{n+1} - 2) \right). \tag{12}$$

Equations (11) and (12) are exactly in the same form as Equation (2) which describes the operation of a threshold gate. The input weights for calculating G_j^{j-n+1} and P_j^{j-n+1} are the same, and the gate thresholds differ by 1.

An example will serve to illustrate the ideas. Consider a 3-bit grouping of the input bits $(a_5, b_5, a_4, b_4, a_3, b_3)$. The group generate signal G_5^3 is 1 if the sum of the inputs is greater than the largest number representable in the

three sum bits (s_5, s_4, s_3), 7. The group propagate P_5^3 signal is 1 if the sum of the inputs is greater than or equal to 7. This can be expressed as:

$$G_5^3 = \mathrm{sgn}(4a_5 + 4b_5 + 2a_4 + 2b_4 + a_3 + b_3 - 8) \quad (13)$$

$$P_5^3 = \mathrm{sgn}(4a_5 + 4b_5 + 2a_4 + 2b_4 + a_3 + b_3 - 7). \quad (14)$$

An expression for calculating G_j^0 may be written by combining the intermediate group generate and group propagate signals in the following way:

$$G_j^0 = G_j^k + P_j^k G_{k-1}^l + P_j^k P_{k-1}^l G_{l-1}^m + P_j^k P_{k-1}^l P_{l-1}^m \ldots P_{x-1}^z G_{z-1}^0. \quad (15)$$

Equation (15) can be interpreted as expressing the partitioning of the w inputs into contiguous blocks in which it is determined where a carry signal is generated and propagated. Such an expression may easily be converted into TL form. This is illustrated by the following example for G_{15}^0, where we partition the 16 bits into 4 groups, and use 4 bit group generate and group propagate signals as follows:

$$\begin{aligned} G_{15}^0 &= G_{15}^{12} + P_{15}^{12} G_{11}^8 + P_{15}^{12} P_{11}^8 G_7^4 + P_{15}^{12} P_{11}^8 P_7^4 G_3^0 \\ &= \mathrm{sgn}\left(8G_{15}^{12} + 4P_{15}^{12} + 4G_{11}^8 + 2P_{11}^8 + 2G_7^4 + P_7^4 + G_3^0 - 8\right). \end{aligned} \quad (16)$$

Finally, the sum bits are computed as follows:

$$\begin{aligned} s_j &= \bar{a}_j \bar{b}_j c_{j-1} \bar{c}_j + \bar{a}_j b_j \bar{c}_{j-1} \bar{c}_j + a_j \bar{b}_j \bar{c}_{j-1} \bar{c}_j + a_j b_j c_{j-1} c_j \\ &= \mathrm{sgn}(a_j + b_j + c_{j-1} - 2c_j - 1) \\ &= \mathrm{sgn}(a_j + b_j + c_{j-1} + 2\bar{c}_j - 3), \end{aligned} \quad (17)$$

where we have used $-c_j = \bar{c}_j - 1$ so that all threshold gate weights are positive. This will be required when we discuss implementation issues in the following Section. It is interesting to note that the TL design discussed above does not require the explicit computation of bit-wise carry generate and carry propagate signals in the first stage of the adder as is required in the traditional CMOS approach. This further contributes to reducing the logic depth. Additionally the final sum bit at each significance is computed by a single gate.

6. LOW DEPTH CARRY LOOKAHEAD ADDERS

By exploiting the parallelism inherent in the computation of carry signals as expressed in Equation (15), we can construct carry lookahead adders of significantly reduced logic depth compared to previous prefix tree approaches.

The 4-bit carry lookahead tree structure is shown in Fig. 5. The cells in the first layer of the adder compute in parallel all four carry signals, and the second layer computes the sum according to Equation (17). This adder assumes a zero carry-input into the least significant position, but this may easily be accomodated as shown in the circuit diagram realisation of this structure in Fig. 6. This figure also shows the capacitance values as multiples of the unit capacitance. A suitable value of the unit capacitance for a 0.25 μm CMOS process is 5 fF. Each of the cells in Fig. 6 consists of one CRTL gate. The cells in each layer are clocked simultaneously.

To measure the power dissipation, the adder was loaded with minimum sized inverters and the input vectors were set to $(a_3\ a_2\ a_1\ a_0) = (0\ 0\ 0\ 0)$ and $(b_3\ b_2\ b_1\ b_0) = (1\ 1\ 1\ 1)$. The c_{-1} was switched from 0 to 2 V at a frequency of 100 MHz and each gate was clocked at 200 MHz. The simulated power dissipation was 640 μW.

Larger adders may be composed as shown for a 16-bit wordlength in Fig. 7. The black cells consist of two CRTL gates and compute GP_j^i signals, the grey cells compute the carry signals G_j^0 and the white cells compute the sum. Only one capacitive input network is required for computing the pair of GP_j^i signals because the input weights for computing group propagate and group generate are the same, and it is shared by the two CRTL gates which have different thresholds. This is one of many possible designs and it can be seen that the adder has a depth of only 4 gates. This is a significant improvement compared to, for example, a conventional 16-bit Brent-Kung adder which has a critical path consisting of 9 gates (7 gates for the prefix-tree, one gate for generating p_i and g_i and finally one XOR gate for computing the sum bits). To achieve bit-level pipelined operation, the input operands (a_j, b_j) must propagate through the CLA because the sum bits s_j are computed from the input operands as well as the two carries (c_{j-1}, c_j). Therefore each cell in the CLA must also include two D-latches, which are not shown in Fig. 7. This would result in a compact and potentially low-power pipelined adder suitable for DSP applications in portable systems. It

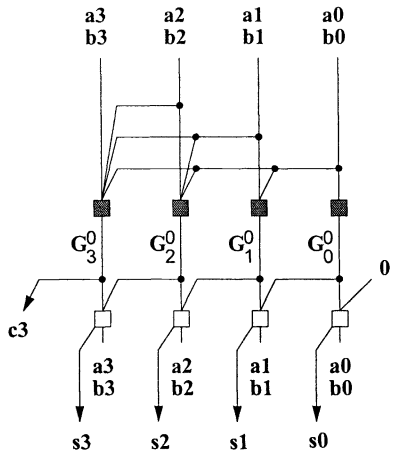

Figure 5. Structure of 4-bit carry lookahead adder with no carry-in

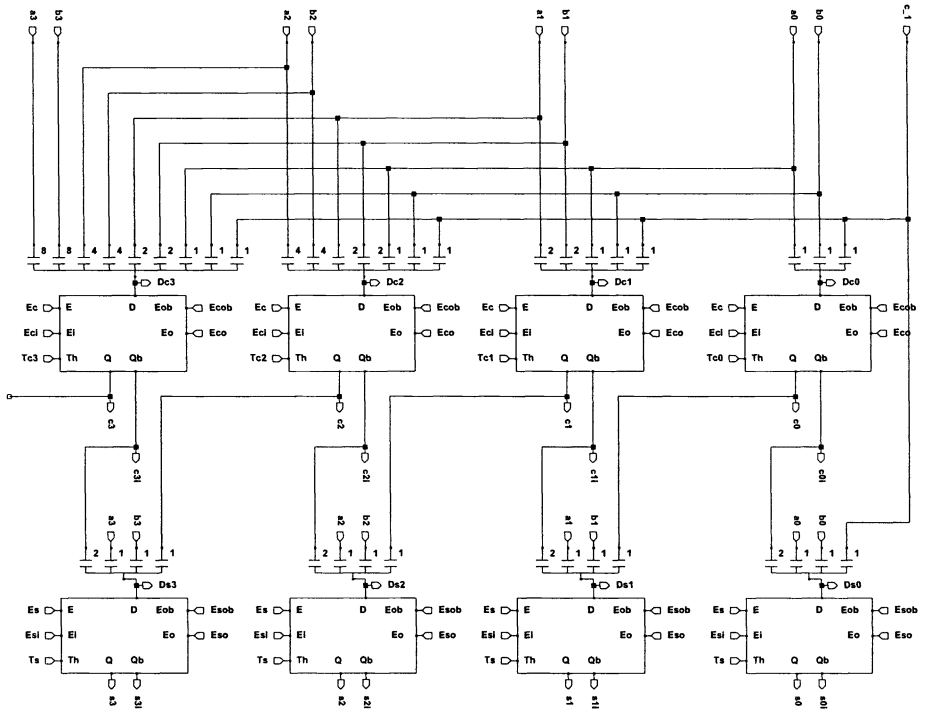

Figure 6. Circuit diagram of simulated 4-bit carry lookahead adder with carry-in (c_{-1}) included

can also be said that the proposed CLA has a number of very desirable properties. The adder consists primarily of only one type of CRTL gate, which means only one gate requires careful design and optimization (in addition to the relatively simple capacitive networks). The regularity of each cell also means that networks of the type shown in Fig. 7 are highly suitable for automated layout generation. The complete characterisation of the performance and power dissipation of CRTL CLA adders, as well as the development of a family of CLAs is the subject of ongoing work.

7. CONCLUSIONS

A new CMOS threshold-logic gate has been proposed. A 20-input majority gate has been designed and simulated using the proposed CRTL structure in order to demonstrate its operation. A comparison with other TL realisations

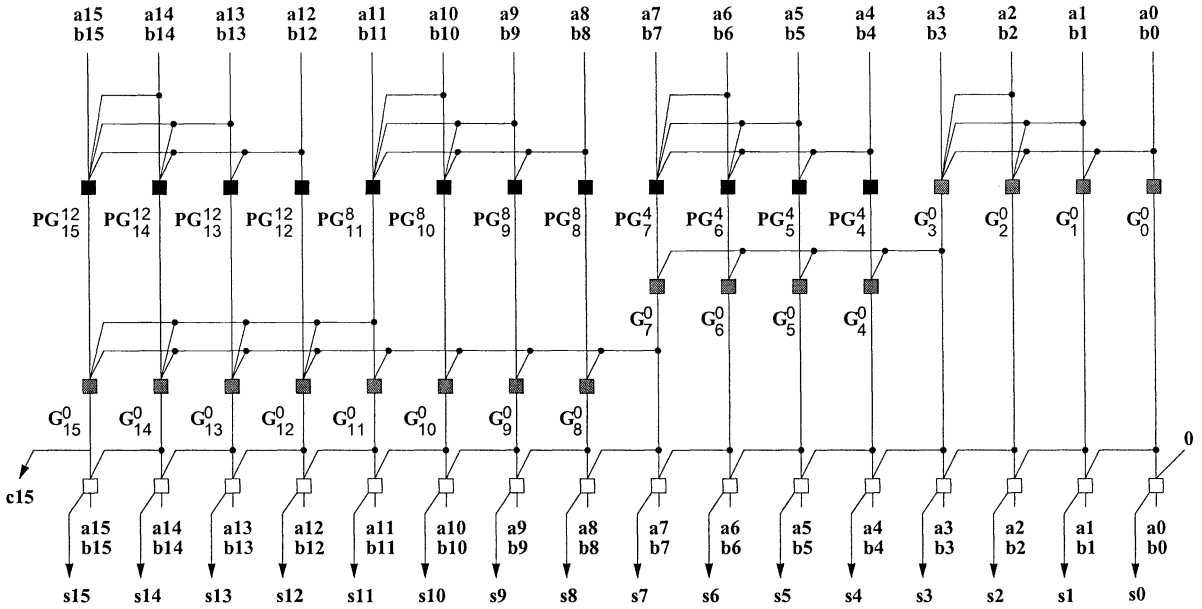

Figure 7. Structure of 16-bit carry lookahead adder with 4-bit grouping and no carry-in

shows that this threshold gate has very low power dissipation. The gate is able to operate at clock frequencies of over 400 MHz, and it is robust under process, supply voltage and temperature variations. A novel carry lookahead addition scheme was also proposed and 4-bit and 16-bit adders designs based on this scheme were presented, demonstrating low depth carry lookahead addition.

ACKNOWLEDGMENTS

The support of the Australian Research Council and the Sir Ross and Sir Keith Smith Fund is gratefully acknowledged. The first author would also like to thank the support of the Institute for Applied Microelectronics at the Universidad de Las Palmas G.C., Spain where he was hosted during 2001 for a period of collaboration.

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*, pp. 320–321, 1995.
2. P. Celinski, S. Al-Sarawi, and D. Abbott, "A delay model for neuron-CMOS and Cpacitive Threshold Logic," in *Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems*, pp. 932–935, (Lebanon), December 2000.
3. H. Huang and T. Wang, "CMOS capacitor coupling logic (C^3L) circuits," in *Proc. of IEEE Asia Pacific Conference on ASIC*, pp. 33–36, 2000.
4. M. Avedillo, J. Quintana, A. Rueda, and E. Jiménez, "Low-power CMOS threshold-logic gate," *IEE Electronics Letters* **31**, pp. 2157–2159, Dec. 1995.
5. 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.
6. S. Muroga, *Threshold Logic and Its Applications*, Wiley, New York, 1971.
7. V. Bohossian, *Neural Logic: Theory and Implementation.* PhD thesis, California Institute of Technology, July 1998.
8. B. Kong, J. Im, Y. Kim, S. Jang, and Y. Jun, "Asynchronous sense differential logic," in *ISSCC Digest of Technical Papers.*, pp. 284–285, 1999.
9. A. Beaumont-Smith and C.-C. Lim, "Parallel prefix adder design," in *Proceedings of the 15th IEEE Symposium on Computer Arithmetic*, (Vail, USA), June 2001.