LOW DEPTH CARRY LOOKAHEAD ADDITION USING CHARGE RECYCLING THRESHOLD LOGIC

Peter Celinski, Said Al-Sarawi, Derek Abbott

Centre for High Performance Integrated Technologies & Systems (CHiPTec), The Department of Electrical and Electronic Engineering, Adelaide University, SA 5005, Australia.
celinski@eleceng.adelaide.edu.au

José F. López

Research Institute for Applied Microelectronics, Universidad de Las Palmas de G.C., 35017-Spain.
lopez@iuma.ulpgc.es

ABSTRACT

The main result of this paper is the development of a low depth carry lookahead addition technique based on threshold logic. Two such adders are designed using the recently proposed charge recycling threshold logic gate. The adders are shown to have a very low logic depth, and significantly reduced area and power dissipation compared to other dynamic CMOS implementations.

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 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].

We begin in Section 2 by giving a brief overview of threshold logic (TL). This is followed by a description of Charge Recycling Threshold Logic (CRTL) [1] and a comparison of the power dissipation of CRTL with other TL gates in Section 3. In Section 4 the proposed carry lookahead addition scheme is presented, and the designs for two novel CLAs based on CRTL are shown 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.

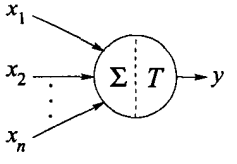

Fig. 1. Threshold Gate Model

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 ith input variable y_i. The output y is given by

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

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

$$y = \text{sgn}\left(\sum_{i=1}^{n} w_i x_i - T\right). \quad (2)$$

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

0-7803-7448-7/02/$17.00 ©2002 IEEE

threshold, and all Boolean functions can be realized by a threshold gate network of depth at most two. 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.

3. CHARGE RECYCLING THRESHOLD LOGIC (CRTL)

We now briefly describe the realization for CMOS threshold gates presented in [1]. Fig. 2 shows the circuit structure which is based on the charge recycling Asynchronous Sense Differential Logic (ASDL) developed by Bai-Sun *et al.* [4] 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.

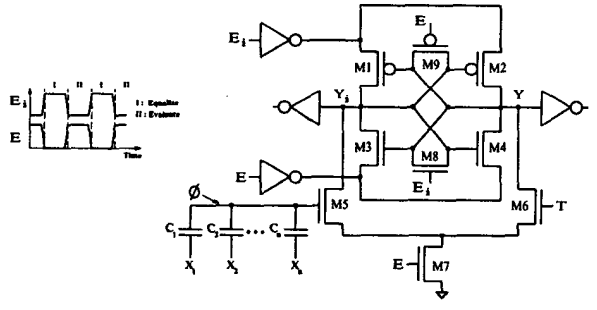

Fig. 2. The proposed CRTL gate structure

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 have high operating speed and typically around 15 to 20% lower power dissipation than other capacitive CMOS TL gate implementations [3] [2]. It was also shown to operate reliably at frequencies over 400 MHz.

4. 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, 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 leads to 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 word length 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^i, propagate, P_j^i, and carry c_j signals at each bit position j for a width w adder as follows:

$$G_j^i = \begin{cases} a_j.b_j, & \text{for } i = j \\ G_j^k + P_j^k.G_{k-1}^i, & \text{for } j \geq k > i \end{cases} \quad (3)$$

$$P_j^i = \begin{cases} a_j + b_j, & \text{for } i = j \\ P_j^k.P_k^i, & \text{for } j \geq k > i, \end{cases} \quad (4)$$

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

$$c_j = G_j^0. \quad (5)$$

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^j and P_i^j, carry lookahead adders are usually built using a parallel prefix-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.

A modified set of the Boolean Equations (3)-(5) 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 threshold logic prefix-cells, that is, prefix-cells which compute group propagate, group generate and carry signals from a large number of input bits.

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. Similarly, 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:

$$G_j^{j-n+1} = \text{sgn}\Big(\sum_{k=j-n+1}^{j} 2^{k-(j-n+1)}(a_k + b_k) - 2^n\Big) \quad (6)$$

$$P_j^{j-n+1} = \text{sgn}\Big(\sum_{k=j-n+1}^{j} 2^{k-(j-n+1)}(a_k + b_k) - 2^n - 1\Big). \quad (7)$$

Equations (6) and (7) 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), which is 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:

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

$$P_5^3 = \text{sgn}(4a_5 + 4b_5 + 2a_4 + 2b_4 + a_3 + b_3 - 7). \quad (9)$$

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

$$G_j^0 = G_j^k + P_j^k G_{k-1}^l + P_j^k P_{k-1}^l G_{l-1}^m + P_j^k P_{k-1}^l P_{l-1}^m \ldots P_{x-1}^z G_{z-1}^0. \quad (10)$$

Equation (10) 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 also be easily 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:

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

Finally, the sum bits are computed from the truth table for addition as follows:

$$\begin{aligned} s_j &= \text{sgn}(a_j + b_j + c_{j-1} - 2c_j - 1) \\ &= \text{sgn}(a_j + b_j + c_{j-1} + 2\bar{c}_j - 3), \end{aligned} \quad (12)$$

where we have used $-c_j = \bar{c}_j - 1$ so that all weights are positive.

5. LOW DEPTH CARRY LOOKAHEAD ADDERS

By exploiting the parallelism inherent in the computation of carry signals as expressed in Equation (10), we can construct carry lookahead adders of significantly reduced logic depth compared to previous prefix-tree approaches.

One possible 16-bit carry lookahead tree structure is shown in Fig. 4. The black cells consist of two CRTL gates and compute GP_j^i signal pairs, the grey cells compute the carry signals G_j^0 and the white cells compute the sum according to Equation (12). 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. 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 would also need to include two D-latches, which are not shown in Fig. 4. This would result in a compact and potentially low-power pipelined adder suitable for DSP applications. In addition, 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. 4 are highly suitable for automated layout generation.

A 4-bit adder was chosen for evaluation by simulation and the circuit diagram with carry-in included is shown in

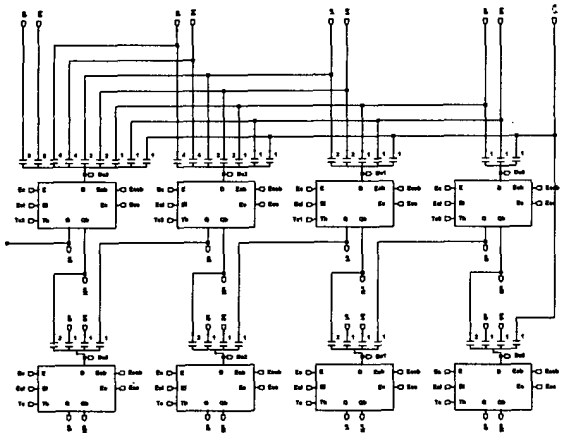

Fig. 3. Circuit diagram of the simulated 4-bit carry lookahead adder with carry-in (c_{-1}) included

Table 1. MODCVS and CRTL 4-bit adder comparison

	MODCVS	CRTL	Reduction
Transistors	176	104	41%
Area	$52\times72\mu m^2$	$43\times53\mu m^2$	39%
Avg. power (@ 100 MHz)	1.9 mW	0.5 mW	74%

Fig. 3. 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) and simulated using HSPICE. The c_{-1} was switched from 0 to 2 V at a frequency of 100 MHz and each gate was clocked at 200 MHz. To perform a comparison of the proposed CRTL design, a 4-bit multi-output differential cascode voltage switch (MODCVS) logic adder [5] was also designed in the same 0.25μm process and simulated under the same switching conditions. The MODVCS design presented in [5] calculates all 4 carries in parallel and is claimed to have low power dissipation, low transistor count and low area compared to previous DCVS logic designs. For this reason it was chosen for comparison with CRTL. Both adders were laid out in the same 0.25μm, 4-metal, 2-poly process. The poly1-poly2 capacitance in this process is 0.8 fF/μm^2, and as was mentioned earlier, the unit capacitance was chosen to be 5fF. The comparison results are shown in Table 1 and demonstrate that CRTL offers significant improvement over the MODCVS design. It is interesting to note that the area of the capacitive networks in the CRTL adder comprise approximately half of the total area of the adder. The reason for this is the relatively low poly1-poly2 capacitance value in the chosen process, and further significant area reduction would be possible by using processes with dedicated capacitive layers such as MIM.

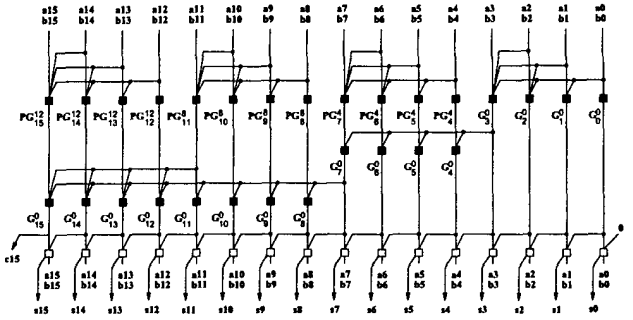

Fig. 4. Structure of a 16-bit carry lookahead adder with 4-bit grouping and no carry-in

6. CONCLUSIONS

A low depth carry lookahead addition technique based on threshold logic was presented. A 4-bit and a 16-bit adder were designed based on this technique using the recently proposed charge recycling threshold logic gate. The proposed addition technique significantly reduces logic depth over the conventional prefix-tree approach. Simulation results also indicate improved area and power dissipation compared to the multi-output differential cascode voltage switch implementation.

Acknowledgments

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

7. REFERENCES

[1] 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.

[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] 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.

[4] B.S. Kong, J.D. Im, Y.C. Kim, S.J. Jang, and Y.H. Jun, "Asynchronous sense-differential logic," in *ISSCC Digest of Technical Papers.*, 1999, pp. 284–285.

[5] G. A. Ruiz, "Compact four bit carry look-ahead CMOS adder in multi-output DCVS logic," *IEE Electronics Letters*, vol. 32, no. 17, pp. 1556–1557, August 1996.