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

Jose 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.

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

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 $i$th 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
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 $\phi$ of $M_5$, and the threshold is set by the gate voltage $T$ of $M_6$. The potential $\phi$ 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](image)

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 ($M_5$-$M_7$) 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, $\phi$, 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 $M_6$, 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, $G_j$, propagate, $P_j$, and carry $c_j$ signals at each bit position $j$ for a width $w$ adder as follows:

$$G_j = \begin{cases} a_j \cdot b_j, & \text{for } i = j \\ G_k + P_j \cdot G_{k-1}, & \text{for } j \geq k > i \end{cases}$$

$$P_j = \begin{cases} a_j + b_j, & \text{for } i = j \\ P_k \cdot P_j, & \text{for } j \geq k > i \end{cases}$$

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:

$$c_j = G_j^0.$$
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 \((G_j^{2-n+1} \text{ and } P_j^{2-n+1})\) 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 \(G_j^{2-n+1}\) 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 \(P_j^{2-n+1}\) 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:

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

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

Equations (6) and (7) are exactly in the same form as Equation (2) which expresses the operation of a threshold gate. The input weights for calculating \(G_j^{2-n+1}\) and \(P_j^{2-n+1}\) 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:

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

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

An expression for calculating \(G_5^0\) may be written by combining the intermediate group generate and group propagate signals in the following way:

\[
G_5^0 = G_5^3 + P_5^3 G_{k-1}^3 + P_5^3 P_{k-1}^3 G_{l-1}^m + P_k^3 P_{k-1}^m P_{l-1}^m \ldots P_{a-1}^m G_{z-1}^0.
\]

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:

\[
G_{15}^0 = G_{15}^3 + P_{15}^3 G_{11}^3 + P_{15}^3 P_{11}^3 G_7^3 + P_{15}^3 P_{11}^3 P_1^3 G_5^0
\]
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.

**6. CONCLUSIONS**

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

**7. REFERENCES**


